

Optimale Partner offener Systeme

Modellierung, Analyse, Synthese

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium

(Dr. rer. nat.)

im Fach Informatik

eingereicht an der

Mathematisch-Naturwissenschaftlichen Fakultät der
Humboldt-Universität zu Berlin

von

Herrn Diplom-Informatiker

Jan Sürmeli

Präsident der Humboldt-Universität zu Berlin:

Prof. Dr. Jan-Hendrik Olbertz

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:

Prof. Dr. Elmar Kulke

Gutachter:

Prof. Dr. Wolfgang Reisig

Prof. Dr. Manfred Droste

PD Dr. Louchka Popova-Zeugmann

Eingereicht am: 15.12.2014

Verteidigt am: 30.04.2015

Zusammenfassung

Heutzutage besteht ein komplexes Software-System häufig aus lose gekoppelten, interagierenden Komponenten. Eine Komponente ist ein *offenes System*, das unabhängig von anderen offenen Systemen entwickelt und später mit diesen *komponiert* wird. Die Komposition $L \oplus R$ zweier offener Systeme L und R kann sich jedoch *inkorrekt* verhalten, beispielsweise verklemmen (die Komponenten warten gegenseitig aufeinander), in eine Endlosschleife geraten oder unbeschränkten Speicherplatz erfordern. Ist $L \oplus R$ dagegen ein *korrektes* System, bezeichnet man L und R als *Partner* voneinander. *Formale Methoden* der Modellierung, Analyse und Synthese ermöglichen die systematische Konstruktion eines korrekten Systems durch Komposition von Partnern.

Die *Kosten*, die ein offenes System L verursacht, variieren in Abhängigkeit von der konkreten Wahl eines Partners. Es ist daher wünschenswert, L nur mit solchen Partnern zu komponieren, welche die Kosten von L beschränken oder sogar minimieren. Ein Partner, der die Kosten von L minimiert, ist ein *optimaler Partner* von L . Ziel dieser Arbeit ist die Erarbeitung von Techniken, die garantieren, dass L nur mit optimalen Partnern komponiert wird.

Dazu entwickeln wir formale Methoden zur Modellierung, Analyse und Synthese kostenbehafteter offener Systeme und ihrer optimalen Partner. Wir präsentieren einen Formalismus zur *Modellierung funktionaler* (d.h. Zustandsübergänge) und *nicht-funktionaler Verhaltenseigenschaften* (d.h. Kosten). In diesem Formalismus definieren wir *Kostenbeschränktheit* und *Optimalität* von Partnern. Darauf aufbauend entwickeln wir formale Methoden zur *Entscheidung der kostenbeschränkten Bedienbarkeit* (d.h. der Existenz kostenbeschränkter Partner), der *Synthese optimaler Partner* und der *endlichen Repräsentation aller optimalen Partner*.

Abstract

Nowadays, a complex software system usually consists of loosely-coupled, interacting components. Such a component is an independently developed *open system* that one *composes* with other open systems. The composition $L \oplus R$ of two open systems L and R can be faulty: For instance, the components deadlock (i.e. mutually wait for each other) or require an unbounded amount of memory. If $L \oplus R$ is *correct*, L and R are called *partners* of each other. *Formal methods* for modeling, analysis and synthesis yield a systematic approach to constructing a correct system by means of composing partners.

The *costs* of executing a given open system L vary based on a chosen partner. Therefore, it is desirable to choose a partner that bounds or even minimizes the costs of executing L . If a partner R minimizes the costs of executing L , then R is an *optimal partner* of L . Our goal is to develop techniques that guarantee the composition of L with optimal partners.

To this end, we develop formal methods of modeling, analysis and synthesis of open systems incorporating costs. We present a formalism to model *functional* aspects (i.e. states and transitions) and *non-functional* aspects (costs) of *behavior*. We define the properties of *cost boundedness* and *cost optimality* for partners in this formalism. Based thereon, we develop formal methods to *decide cost bounded controllability* (i.e. the existence of cost bounded partners), to *synthesize optimal partners*, and to *finitely represent the set of all optimal partners*.

Inhaltsverzeichnis

I	Einleitung	1
1	Über diese Arbeit	3
1.1	Hintergrund	3
1.2	Problemstellung	5
1.3	Resultate	10
1.4	Aufbau der Arbeit	11
1.5	Lösungsansatz: Analyse mit Repräsentanten	12
2	Mathematische Grundlagen	15
2.1	Natürliche und reelle Zahlen	15
2.2	Wörter und Sprachen	15
2.3	Funktionen, Familien und Multimengen	15
2.4	Ordnungstheoretische Grundlagen	18
2.5	Partiell vollständige Strukturen	21
2.6	Automaten	25
2.7	Petrinetze	26
3	Gewichtete Netze	39
3.1	Syntax gewichteter Netze	39
3.2	Perspektiven	41
3.3	Kostenmodelle	46
3.4	Semantik gewichteter Netze	66
3.5	Kosteneindeutigkeit	70
3.6	Anforderungen	86
3.7	Präferenzen	91
3.8	Diskussion	92
II	Analyse geschlossener Systeme	99
4	Repräsentanten geschlossener Systeme	101
4.1	Perspektiven-Repräsentanten	101
4.2	Wert-Repräsentanten	109
4.3	Schranken-Repräsentanten	113
4.4	Diskussion	115

5	Analyse auf Basis von Repräsentanten	117
5.1	Berechnung der Wert-Semantik mit Repräsentanten	117
5.2	Entscheidung der k-Kostenbeschränktheit mit Repräsentanten	120
5.3	Entscheidung der Kostenbeschränktheit mit Repräsentanten	123
5.4	Diskussion	126
6	Exkurs: Analyse von Multihemiring-Automaten	129
III	Analyse und Synthese offener Systeme	135
7	Optimale Partner offener Systeme	137
7.1	Komposition	137
7.2	Partner offener Systeme	140
7.3	Optimale Partner offener Systeme	153
7.4	Diskussion	161
8	Analyse- und Syntheseprobleme	165
9	Repräsentanten offener Systeme	167
9.1	Kompositionale Repräsentanten	167
9.2	Partner-Repräsentanten	173
9.3	Diskussion	179
10	Analyse und Synthese auf Basis von Repräsentanten	181
10.1	Entscheidung der Bedienbarkeit bezüglich Kostenbeschränktheit	181
10.2	Partnersynthese und Repräsentation für Kostenbeschränktheit	185
10.3	Diskussion	187
11	Kostendiskriminanten offener Systeme	189
11.1	Kostendiskriminanten für beliebige Kostenmodelle	189
11.2	Konstruktion von Kostendiskriminanten	190
11.3	Diskussion	194
12	Analyse und Synthese auf Basis von Kostendiskriminanten	195
12.1	Entscheidung der \mathbb{SB}_μ^π -Bedienbarkeit	195
12.2	Partnersynthese für schwache Terminierung und Kostenbeschränktheit	197
12.3	Kostenoptimale Bedienbarkeit bezüglich schwacher Terminierung	198
12.4	Synthese und Repräsentation kostenoptimaler Partner	198
12.5	Diskussion	202

IV	Schluss	205
13	Fallstudie	207
13.1	Implementation	207
13.2	Evaluation	209
14	Fazit	217
14.1	Zusammenfassung der Ergebnisse	217
14.2	Einschränkungen und offene Probleme	220
14.3	Zukünftige Arbeiten	223
	Appendix	225
	Literatur	227
	Glossar	235
	Symbole	237

Teil I

Einleitung

In diesem ersten Teil beschreiben wir die vorliegende Arbeit in Kapitel 1, S. 3, wiederholen die von uns benötigten mathematischen Grundlagen in Kapitel 2, S. 15 und stellen *gewichtete Netze* in Kapitel 3, S. 39 als Formalismus zur Modellierung offener Systeme vor.

1 Über diese Arbeit

In diesem Kapitel beschreiben wir den Hintergrund der Arbeit Abschnitt 1.1, formulieren die Problemstellung in Abschnitt 1.2, S. 5, präsentieren abstrakt unsere Resultate in Abschnitt 1.3, S. 10, beschreiben den Aufbau der Arbeit in Abschnitt 1.4, S. 11 und stellen unsere zentrale Lösungstechnik in Abschnitt 1.5, S. 12 vor.

1.1 Hintergrund

Computergestützte Systeme sind heute grundlegender Bestandteil der modernen Gesellschaft. Die Entwicklung computergestützter interner und globaler Netzwerke sowie die weite Verbreitung netzwerktauglicher Geräte und eingebetteter Systeme [Lee02] ermöglicht die Koordination von und den Datenaustausch zwischen einer Vielzahl von Menschen und Geräten. In diesem Rahmen wird es zunehmend schwieriger die *Korrektheit* eines Systems zu beschreiben und zu garantieren [Fro04]. Gleichzeitig können inkorrekte Systeme zu hohen Vermögens-, Personen- und Sachschäden führen. *Formale Methoden* der Spezifikation und Modellierung beschreiben Systeme und ihre Eigenschaften als mathematische Objekte mit wohldefinierter Syntax und Semantik, beispielsweise als Kripke-Strukturen [CGP01], Formeln temporaler Logik [Pnu77], abstrakte Zustandsmaschinen [Guro0], Petrinetze [Rei13] oder Terme einer Prozessalgebra [Bae05]. Dies ermöglicht den Einsatz von Techniken zur Analyse und Synthese: Bei der Analyse überprüft man die Korrektheit eines gegebenen Systems, bei der Synthese konstruiert man ein korrektes System. Die Reduktion von *Kosten* durch *Optimierung* von Systemen ist ein weiteres Ziel neben der Korrektheit. Dabei sind Kosten als abstrakter Begriff zu verstehen, der unter anderem für monetäre Kosten, Laufzeit, Speicherbedarf oder Energieverbrauch stehen kann. Wie bei der Korrektheit stehen hier ebenfalls formale Methoden zur Verfügung, wie beispielsweise Compiler-Techniken, die ein Programm optimieren, ohne seine Semantik zu verändern.

Die zunehmende Vernetzung und Komplexität von Systemen brachte neue Paradigmen für die Entwicklung von Systeme hervor, wie die *Objekt-Orientierung* [Mey88; Rum+91; Gam+94], *Komponenten-Orientierung* [McI68; Szy02], *Service-Orientierung* [Pap08; Hul+03; Got00] und *Cloud-Computing* [Arm+10; MG11]. Diesen Paradigmen zufolge sind große Systeme aus kleineren, interagierenden *Komponenten* aufgebaut. Dabei sollen die Komponenten möglichst *lose*

gekoppelt sein, das heißt, eine Komponente wird von vornherein unabhängig von anderen Komponenten entwickelt, um später Teil mehrerer, verteilter Systeme zu sein. Erstellt man ein neues System, greift man also auf bestehende Komponenten zurück oder entwickelt, falls notwendig, neue Komponenten.

Die unabhängig voneinander entwickelten Komponenten sollen später *korrekt miteinander interagieren*. Dies ist jedoch im Allgemeinen nicht der Fall: Ein aus zwei Komponenten L und R komponiertes System $L \oplus R$ kann beispielsweise verklemmen (L und R warten gegenseitig aufeinander), in eine Endlosschleife geraten, den Bedarf an Speicher unbeschränkt wachsen lassen oder unbeschränkt hohe Kosten verursachen. Interagieren die Komponenten eines Systems nicht korrekt, ist das Gesamtsystem inkorrekt. Im Falle interagierender Komponenten ist die Erkennung und Korrektur inkorrektur Zustände zur Laufzeit besonders schwierig: Beispielsweise bedarf die Erkennung und Auflösung einer Verklemmung oder einer Endlosschleife potenziell einer vertrauenswürdigen Drittpartei (englisch: Trusted Third Party, kurz: TTP), die genaue Einsicht in den jeweiligen Zustand aller beteiligten Komponenten hat. Dies ist kostenaufwändig und stellt ein Risiko für die beteiligten Parteien da. Um die Korrektheit eines Systems zu garantieren, ist es daher notwendig bei der Entwicklung bereits sicherzustellen, dass die Komponenten zur Laufzeit korrekt miteinander interagieren. Dies kann man durch die geschickte Auswahl der Komponenten, Adaption bestehender Komponenten oder die Entwicklung neuer Komponenten erreichen. Dafür ist es essenziell zu wissen, welche Komponenten korrekt miteinander interagieren, um nur solche Komponenten miteinander zu komponieren.

Folgt man dem Prinzip der losen Kopplung, bedeutet *Optimierung* eines Systems die Auswahl und Erstellung genau solcher Komponenten, die die Kosten minimieren. Da die Komponenten miteinander interagieren, hängen die von einer Komponente verursachten Kosten von der Auswahl der anderen Komponenten ab. Beispielsweise kann eine Komponente L mit einer Komponente R niedrige und mit einer anderen Komponente S hohe Laufzeit aufweisen. Gleichzeitig ist es möglich, dass der Energieverbrauch von L mit R höher ist als mit S . Die Kosten eines Systems können zudem für unterschiedliche Fälle optimiert werden, wie zum Beispiel dem *Best*, *Average* oder *Worst Case*. Dabei ist die Auswahl des Falls entscheidend: Die Komponente L könnte zum Beispiel im Best Case mit R niedrigere Kosten als mit S und im Worst Case mit S niedrigere Kosten als mit R verursachen. Daher ist es schwierig, eine einzelne Komponente L zu optimieren, da ihre Kosten in Abhängigkeit der Auswahl der anderen Komponenten variiert. Stattdessen kann man ein System optimieren, indem man es aus genau solchen Komponenten konstruiert, die gemeinsam die Kosten minimieren. Dazu kann man optimal interagierende Komponenten auswählen, einzelne Komponenten optimal adaptieren oder neue Komponenten konstruieren, die per Konstruktion die Kosten minimieren. Möchte man auf diese Weise ein optimales System aus Komponenten

entwickeln, ist es daher essenziell, die von einer Komponente in Abhängigkeit anderer Komponenten verursachten Kosten zu kennen.

In dieser Arbeit folgen wir der formalen Herangehensweise, die *Partner* (auch: *Controller*, *Strategien*) einer Komponente zu betrachten: Ist das System $L \oplus R$ korrekt, sind die Komponenten L und R *Partner* — Partner interagieren also miteinander stets korrekt. Auf den ersten Blick erscheint es als eine Einschränkung, nur die Komposition zweier Komponenten zu betrachten: Ein System ist im Allgemeinen aus mehr als nur zwei Komponenten aufgebaut. Wir können jedoch den Fall der Komposition endlich vieler Komponenten S_1, \dots, S_n als einen Spezialfall der Komposition zweier Komponenten L und R ansehen, indem wir S_1 als L und die Komposition der Komponenten S_2, \dots, S_n als R auffassen. Die *Menge aller Partner* von L charakterisiert also genau diejenigen Komponenten, mit denen L zu einem korrekten System komponiert werden kann, und beschreibt somit präzise die Semantik von L bezüglich der Komposition. Betrachtet man die Partner einer Komponente, existieren zahlreiche Techniken zur Modellierung, Analyse und Synthese. Dies ermöglicht zum Beispiel die Überprüfung, ob gegebene Komponenten L und R Partner sind, die Auswahl eines Partners R für eine gegebene Komponente L aus einer Menge verfügbarer Komponenten oder die Synthese eines Partners R für eine gegebene Komponente L . Wir betrachten in dieser Arbeit zusätzlich zur Korrektheit die Kosten von Komponenten. Auf dieser Basis prägen wir den Begriff des *optimalen Partners*: Wenn L und R Partner sind, und R die Kosten von L minimiert, ist R ein *optimaler Partner* von L .

Bisher haben wir stets die Unterscheidung zwischen einem System und seinen Komponenten gemacht: L und R sind Komponenten des Systems $L \oplus R$. Wir können jedoch die Komponenten L und R jeweils wieder als Systeme auffassen. Folglich werden Systeme aus Systemen komponiert. Daraus entsteht der Begriff des *offenen Systems*: Ein System heißt offen, falls es eine *Schnittstelle* zur Interaktion mit anderen Systemen aufweist — ansonsten heißt es *geschlossen*. Systeme beider Formen können wir zu neuen Systemen komponieren. Dabei verschiebt die Komposition stets die Systemgrenze: Aus zwei Systemen L und R entsteht ein neues System $L \oplus R$. Sind L und R offen, interagieren L und R in $L \oplus R$. Das System $L \oplus R$ kann selbst wieder offen oder geschlossen sein.

1.2 Problemstellung

In dieser Arbeit betrachten wir ganz allgemein folgendes Problem:

Wie können wir optimale Partner offener Systeme modellieren, analysieren und synthetisieren?

Wir beschreiben die grundlegenden Begriffe dieser Arbeit in Abschnitt 1.2.1 und die Problemdimensionen in Abschnitt 1.2.2. Schließlich formulieren wir die in dieser Arbeit betrachteten Forschungsfragen in Abschnitt 1.2.3, S. 8.

1.2.1 Zentrale Begriffe

Wir beschreiben in diesem Abschnitt abstrakt die zentralen Begriffe dieser Arbeit. Abbildung 1 zeigt die Halbordnung der Begriffe in Form eines Hasse-Diagramms.

Ein *offenes System* L interagiert mit anderen offenen Systemen. Wir *komponieren* dazu L mit einem anderen offenen System R zu einem neuen offenen System $L \oplus R$.

Eine *Anforderung* ρ beschreibt eine Teilmenge offener Systeme, die ρ genügen. Aus den Begriffen der Komposition und der Anforderung bilden wir den Begriff des *Partners* eines offenen Systems, der stets mit einer konkreten Anforderung ρ parametrisiert ist: Wenn $L \oplus R$ ein Element von ρ ist, dann ist R ein *Partner* von L , und $L \oplus R$ ist eine *Partnerschaft* von L .

Eine *Präferenz* \succsim ordnet offene Systeme. Eine Präferenz ordnet damit auch die Partnerschaften von L . Anhand der Begriffe der Präferenz und der Partnerschaft definieren wir den Begriff des *optimalen Partners*: Ist eine Partnerschaft $L \oplus R$ von L *optimal* in der Menge der Partnerschaften von L , ist R ein *optimaler Partner* von L . Optimalität bedeutet dabei, dass für alle Partnerschaften $L \oplus R'$ gilt $L \oplus R' \not\succsim L \oplus R$.

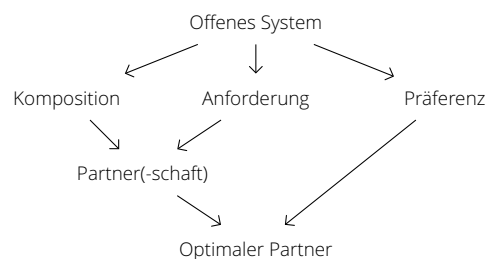


Abbildung 1 Halbordnung der zentralen behandelten Begriffe. „ $A \rightarrow B$ “ bedeutet „ B basiert auf A “.

1.2.2 Problemdimensionen

In dieser Arbeit betrachten wir unterschiedliche Kombinationen aus einer Anforderung ρ und einer Präferenz \succsim . Anforderungen sowie Präferenzen betreffen unterschiedliche Aspekte eines offenen Systems L [Pap07]. Wir unter-

scheiden dabei zunächst *Struktur* und *Verhalten*. Verhaltensaspekte unterteilen wir erneut in *funktionale* Aspekte und *nicht-funktionale* Aspekte.

Strukturelle Aspekte. Die für uns relevanten strukturellen Aspekte eines offenen Systems L sind seine *innere Struktur* sowie seine *Schnittstelle*. Die *innere Struktur* von L beschreibt die lokalen Zustände und legt den *Anfangszustand* fest. Die *Schnittstelle* von L determiniert die strukturellen Grenzen für die Interaktion von L mit anderen offenen Systemen, indem sie auf einem fest gewählten Abstraktionsgrad die Nachrichten festlegt, die L senden oder empfangen kann.

Funktionale Verhaltensaspekte. Wir betrachten vor allem die grundlegenden Begriffe des *Ablaufs* sowie der *Erreichbarkeit* eines Zustands in einem offenen System L . Ein *Ablauf* r von L ist eine Folge erlaubter Zustandsübergänge, überführt also L von einem Zustand p in einen anderen Zustand q . Wenn p der Anfangszustand ist, dann ist q ein *erreichbarer* Zustand in L .

Nicht-funktionale Verhaltensaspekte. Wir betrachten die *Kosten* einzelner Zustandsübergänge sowie die Aggregation von Kosten zu Kosten von Abläufen und Systemen. Wir verstehen dabei Kosten als einen abstrakten Begriff, der beispielsweise für monetäre Kosten, Energieverbrauch, Zuverlässigkeit oder Zeitverbrauch stehen kann. Wir nehmen für die Betrachtung der Kosten von L stets eine *Perspektive* sowie ein *Kostenmodell* an. Eine Perspektive legt fest, welche Abläufe betrachtet werden sollen; ein Kostenmodell definiert die Operationen für die Aggregation von Kosten.

Mithilfe dieser Aspekte lassen sich konkrete Anforderungen und Präferenzen kategorisieren. Wir beschreiben abstrakt die in dieser Arbeit betrachteten Anforderungen und Präferenzen. Abbildung 2 zeigt ihre Kategorisierung — neu in dieser Arbeit eingeführte Begriffe sind mit einem Stern (★) markiert.

Anforderungen an ein offenes System S

Geschlossen. Die Schnittstelle von S ist leer. Diese eher technische Anforderung wird von Kompositionen offener Systeme erfüllt, die ihre Schnittstellen gegenseitig ergänzen.

Beschränkt. In S sind nur endlich viele Zustände erreichbar. Die Beschränktheit von Petrinetzen ist eine wohlbekannte Eigenschaft, Details finden sich beispielsweise bei Reisig [Rei13].

Schwach terminierend. (1) Jeder erreichbare Zustand von S gehört zu einer vordefinierten Menge von *Rahmenzuständen*, und (2) aus jedem erreichbaren Zustand von S ist ein Zustand aus einer vordefinierten Menge

von *Zielzuständen* erreichbar. Schwache Terminierung ist eng verwandt mit der aus der Analyse von Geschäftsprozessen bekannten Begriffsfamilie der *Soundness* [Aal+11]: Schwache Terminierung generalisiert *Weak Soundness* von Martens [Mar05], ist jedoch schwächer als *Classical Soundness* von Aalst [Aal97].

k-Kostenbeschränkt. Die Kosten von S liegen unter einer gegebenen Schranke k [Sür12].

Kostenbeschränkt. S ist k -kostenbeschränkt für wenigstens ein k [Sür12].

Präferenzen auf offenen Systemen S und T

Kostengünstiger. Die Kosten von S sind niedriger als die Kosten von T [ST13].

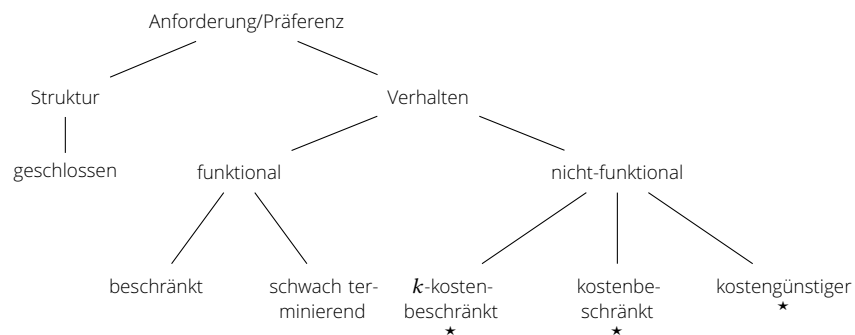


Abbildung 2 Kategorisierung der Anforderungen an Systeme und Präferenzen auf Systemen. In dieser Arbeit neu eingeführte Anforderungen und Präferenzen sind mit einem ★ gekennzeichnet.

1.2.3 Problemstellungen und Forschungsfragen

In der Literatur tauchen eine Reihe von Problemen im Umgang mit offenen Systemen auf, die wir im folgenden kurz beschreiben. Dabei sei stets ein offenes System L , eine Anforderung ρ sowie eine Präferenz \succsim fest gegeben.

Korrektheit bzw. Optimalität. Gegeben: Ein offenes System R . Aufgabe: Entscheide, ob R ein Partner bzw. optimaler Partner von L ist.

Bedienbarkeit [RW87; Cie+88; KMoo; Wol09]. Aufgabe: Entscheide die Existenz wenigstens eines Partner R von L .

Partnersynthese [PR90; KMoo; AV03; Blo+09; Ben+10]. Aufgabe: Konstruiere einen optimalen Partner R von L .

Repräsentation [Loh+08]. Aufgabe: Konstruiere eine endliche Repräsentation aller optimaler Partner.

Adaptersynthese [Ben+05; BBCo5; DSWo6]. Gegeben: Ein offenes System R . Aufgabe: Konstruiere einen optimalen Partner S für $L \oplus R$.

Konfiguration [La 08; Aal+10; Got+09]. Gegeben: L ist *konfigurierbar* [La 08]. Aufgabe: Berechne eine *Konfiguration* C von L , so dass die Anwendung von C auf L der Anforderung ρ genügt und optimal bezüglich \succsim ist.

Discovery [Don+04; Mar05; Sap+06]. Gegeben: Eine endliche Menge \mathcal{R} offener Systeme. Aufgabe: Finde einen optimalen Partner von L in \mathcal{R} .

Austauschbarkeit [Aal+08]. Gegeben: Ein offenes System M . Aufgabe: Entscheide, ob jeder optimale Partner von L auch ein optimaler Partner von M ist.

Wir können viele dieser Probleme auf die Kernprobleme der Partnersynthese und der Repräsentation reduzieren, wie wir in Abbildung 3 darstellen. So lässt sich beispielsweise das Problem der Adaptersynthese wie in [GMW12] beschrieben auf das Problem der Partnersynthese reduzieren. Die drei grundsätzlichen Problemstellungen für offene Systeme sind folglich Bedienbarkeit, Partnersynthese und Repräsentation.

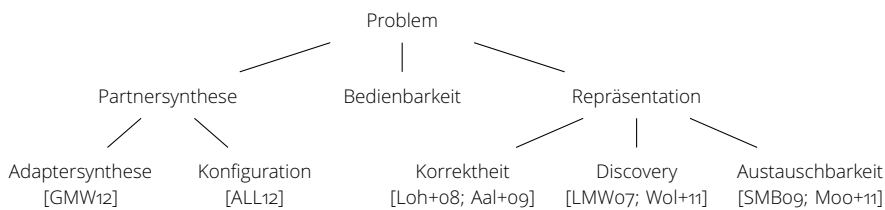


Abbildung 3 Übersicht über die behandelten Probleme. Von oben nach unten werden die Probleme spezieller

Vernachlässigen wir den Aspekt der Optimalität und beschränken wir uns auf Anforderungen, die nur strukturelle und funktionale Verhaltensaspekte berühren, existieren für diese Probleme zahlreiche Lösungen in der Literatur. In dieser Arbeit betrachten wir die folgenden zwei neuen Aspekte:

- (1) Kosten als nicht-funktionale Verhaltensaspekte offener Systeme
- (2) Optimalität von Partnern

Auf dieser Basis formulieren wir unsere Forschungsfragen:

- (1) Wie können wir strukturelle Aspekte gemeinsam mit funktionalen und nicht-funktionalen Verhaltensaspekten formal modellieren?
- (2) Wie können wir Korrektheit und Optimalität offener Systeme bezüglich nicht-funktionaler Verhaltensaspekte modellieren?

- (3) Wie können wir die Probleme der Korrektheit, Bedienbarkeit, Synthese und Repräsentation für Anforderungen und Präferenzen lösen, die strukturelle, funktionale und nicht-funktionale Aspekte berühren?

In Abschnitt 1.3 beschreiben wir abstrakt unsere Resultate zu diesen drei Fragen.

1.3 Resultate

Im Folgenden fassen wir abstrakt die Resultate dieser Arbeit zu den in Abschnitt 1.2, S. 5 gestellten Forschungsfragen.

1.3.1 Formale Modellierung offener Systeme mit gewichteten Netzen

Wir schlagen *gewichtete Netze* in Kapitel 3, S. 39 als Formalismus zur gemeinsamen Modellierung struktureller, funktionaler und nicht-funktionaler Verhaltensaspekte vor. Ein gewichtetes Netz ist ein *Petrinetz*, in der jede Transition Kosten aufweist. Wir definieren zwei parametrisierte Semantiken, die *Wert-* und die *Wortsemantik*. Parameter sind dabei stets eine *Perspektive*, die das zu betrachtende Verhalten festlegt, und ein *Kostenmodell*, das die Aggregation der Kosten definiert.

1.3.2 Formale Modellierung der Korrektheit und Optimalität bezüglich nicht-funktionaler Verhaltensaspekte

Wir schlagen *Anforderungen* in Abschnitt 3.6, S. 86 und *Präferenzen* in Abschnitt 3.7, S. 91 geschlossener Systeme vor, um offene Systeme miteinander zu klassifizieren. Wir leiten die Anforderung der *Kostenbeschränktheit* und die Präferenz *kostengünstiger* aus Perspektiven und Kostenmodellen her, die nicht-funktionale Aspekte des Verhaltens betreffen. Die vorgeschlagenen Konzepte erlauben die Modellierung globaler Korrektheit sowie lokaler und globaler Optimalität.

1.3.3 Analyse und Synthese offener Systeme bezüglich nicht-funktionaler Verhaltensaspekte

Wir entwickeln Methoden zur Analyse und Synthese offener Systeme. Dabei betrachten wir die Probleme der Korrektheit, Bedienbarkeit, Partnersynthese und Repräsentation bezüglich Kostenbeschränktheit und Kostenoptimalität für Teilklassen gewichteter Netze, Perspektiven und Kostenmodellen. Dabei

betrachten wir nicht-funktionale Anforderungen und Präferenzen für das Verhalten allein und in Kombination mit bekannten funktionalen Anforderungen an das Verhalten. Wir betrachten in Teil II, S. 101 zunächst geschlossene Systeme als Grundlage für die Analyse und Synthese offener Systeme in Teil III, S. 137.

1.4 Aufbau der Arbeit

In diesem Abschnitt beschreiben wir den Aufbau der Arbeit. Die Arbeit besteht aus fünf Teilen:

- Teil I, S. 3 leitet die Arbeit ein und stellt die *Modellierung offener Systeme mit gewichteten Netzen* vor: Kapitel 1, S. 3 beschreibt die Arbeit, Kapitel 2, S. 15 beschreibt die notwendigen mathematischen Grundlagen für diese Arbeit und Kapitel 3, S. 39 führt *gewichtete Netze* ein.
- Teil II, S. 101 behandelt die *Analyse geschlossener Systeme*: Kapitel 4, S. 101 führt *Repräsentanten geschlossener Systeme* ein und stellt Methoden zur Konstruktion solcher Repräsentanten vor, Kapitel 5, S. 117 beschreibt die Anwendung der Repräsentanten in der Analyse, Kapitel 6, S. 129 ist ein Exkurs in die Theorie gewichteter Automaten.
- Teil III, S. 137 ist der *Analyse und Synthese offener Systeme* gewidmet: Kapitel 7, S. 137 behandelt die *Komposition* offener Systeme und führt schließlich den Begriff des *optimalen Partners* ein. Kapitel 8, S. 165 beschreibt darauf basierend die Analyse- und Syntheseprobleme, die in dieser Arbeit gelöst werden. Kapitel 9, S. 167 stellt *Repräsentanten* als unseren ersten Lösungsansatz vor, Kapitel 10, S. 181 beschreibt die Anwendung von Repräsentanten in Analyse und Synthese. Kapitel 11, S. 189 beschreibt *Kostendiskriminanten offener Systeme* als zweiten Lösungsansatz und Kapitel 12, S. 195 beschreiben die Anwendung von Kostendiskriminanten in Analyse und Synthese.
- Teil IV, S. 207 bildet den inhaltlichen Schluss der Arbeit: Kapitel 13, S. 207 bildet eine erste Evaluierung der in dieser Arbeit entwickelten Analyse- und Synthesemethode, Kapitel 14, S. 217 zieht ein Fazit über die Arbeit.
- Der Epilog ab S. 227 beinhaltet die Referenzen, ein Glossar und ein Symbolverzeichnis.

1.5 Lösungsansatz: Analyse mit Repräsentanten

Der zentrale Lösungsansatz in dieser Arbeit ist die Analyse mithilfe von *Repräsentanten*. Dazu betrachten wir zunächst ein Analyseproblem φ auf einem Universum \mathcal{U} . Dieses induziert eine Äquivalenzrelation \sim_φ auf \mathcal{U} : Es gelte genau dann $L \sim_\varphi M$ für $L, M \in \mathcal{U}$, wenn das Ergebnis für L und M bezüglich φ gleich ist. Entsprechend entsteht eine Menge $\mathcal{U}/_\varphi = \{[L]_\varphi \mid L \in \mathcal{U}\}$ von Äquivalenzklassen.

Es ist eine typische Praxis, ein neues Problem φ auf ein bereits gelöstes Problem ψ zu reduzieren. Wir verfolgen diesen Ansatz ebenfalls. Dazu identifizieren wir zunächst solche Elemente L aus \mathcal{U} , für die sich Ergebnisse bezüglich ψ direkt auf Ergebnisse bezüglich φ übertragen. Formaler nehmen wir eine berechenbare Funktion $r: \mathcal{U}/_\psi \rightarrow \mathcal{U}/_\varphi$ an, die jede Äquivalenzklasse bezüglich ψ auf eine Äquivalenzklasse bezüglich φ überträgt. Dann ist ein $L \in \mathcal{U}$ genau dann ein $\langle \psi, r, \varphi \rangle$ -*Repräsentant*, wenn $r([L]_\psi) = [L]_\varphi$. Diese Zusammenhänge verbildlichen wir in Abbildung 4.

Beispiel 1

Es ist beispielsweise wohlbekannt, dass wir aus einer quadratischen Funktionsvorschrift in Scheitelpunktsform leicht den Scheitelpunkt ablesen können, indem wir bestimmte Subterme untersuchen. Folglich sind Funktionsvorschriften in Scheitelpunktsform Repräsentanten. Formal gesprochen definieren wir ψ als das Problem, die Koordinaten direkt abzulesen, falls dies möglich ist. Dabei entsteht für jedes Paar $\langle x, y \rangle$ von Koordinaten eine entsprechende Äquivalenzklasse sowie eine Äquivalenzklasse \perp für diejenigen Vorschriften, aus denen der Scheitelpunkt nicht direkt abgelesen werden kann. Analog definieren wir φ als das Problem, die Koordinaten zu berechnen. Dabei entsteht für jedes Paar $\langle x, y \rangle$ genau eine Äquivalenzklasse. Bilde jetzt r die Äquivalenzklassen so ab, dass jede $\langle x, y \rangle$ -Äquivalenzklasse auf sich selbst abbildet und \perp auf eine beliebig gewählte Äquivalenzklasse $\langle x^*, y^* \rangle$ abbildet. Dann sind all solche Funktionsvorschriften $\langle \psi, r, \varphi \rangle$ -Repräsentanten, die in Scheitelpunktsform vorliegen.^a

^aTechnisch gesehen sind außerdem diejenigen Funktionsvorschriften $\langle \psi, r, \varphi \rangle$ -Repräsentanten, die nicht in Scheitelpunktsform vorliegen und die den Scheitelpunkt $\langle x^*, y^* \rangle$ haben.

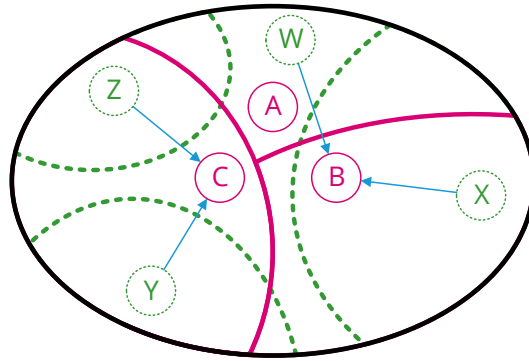


Abbildung 4 *Repräsentanten:* Gegeben seien zwei Probleme φ (Äquivalenzklassen A, B und C) und ψ (Äquivalenzklassen W, X, Y und Z) sowie die Abbildung $r = \{W \mapsto B, X \mapsto B, Y \mapsto C, Z \mapsto C\}$. Die Elemente aus $B \cap \bigcup r^{-1}(B) = B \cap (W \cup X)$ und $C \cap \bigcup r^{-1}(C) = C \cap (Y \cup Z)$ sind Repräsentanten. Die Äquivalenzklasse A enthält keine Repräsentanten, da $A \cap \bigcup r^{-1}(A) = \emptyset$.

Im Allgemeinen ist nicht jedes Element $L \in \mathcal{U}$ ein $\langle \psi, r, \varphi \rangle$ -Repräsentant. Daher verfolgen wir den folgenden Lösungsansatz, den wir in Abbildung 5 veranschaulichen:

- (1) Überführe L in einen $\langle \psi, r, \varphi \rangle$ -Repräsentanten M mit $L \sim_{\varphi} M$.
- (2) Löse ψ für L .
- (3) Übertrage das Ergebnis mithilfe von r .

Im Beispiel der quadratischen Funktionsvorschriften können wir beispielsweise jede Funktionsvorschrift zunächst in Scheitelpunktsform übertragen und anschließend ablesen.

Dieser Lösungsansatz liefert im Allgemeinen nicht das effizienteste Verfahren zur Lösung des Problems φ . Es ist jedoch möglicherweise einfacher die Korrektheit des Verfahrens zu zeigen, falls für das für ψ bekannte Verfahren bereits die Korrektheit gezeigt wurde: In diesem Fall muss zweierlei gezeigt werden:

- (1) Die Korrektheit der Überführung.
- (2) Die Korrektheit der Übertragung.

Wir nehmen an, dass diese beiden Schritte leichter sind als die Entwicklung eines neuen Verfahrens, für das zusätzlich ein Korrektheitsbeweis geführt werden muss.

Nehmen wir auf \mathcal{U} zusätzlich eine binäre Operation \oplus an, können wir den Begriff eines *kompositionalen Repräsentanten* bilden: Ein Element $L \in \mathcal{U}$ ist

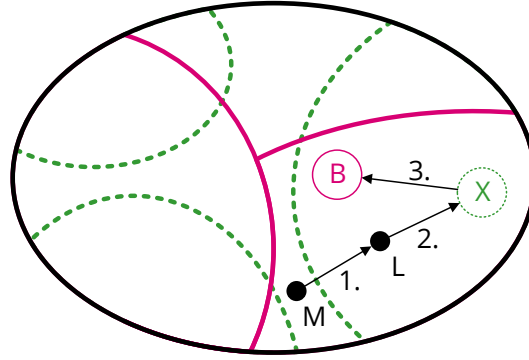


Abbildung 5 *Analyse mit Repräsentanten.* Seien φ , ψ und r wie in Abbildung 4, S. 13 gegeben. Wir analysieren $M \in B$, indem wir M in einen Repräsentanten $L \in B$ überführen. Anschließend lösen wir ψ für L und verwenden r , um das Ergebnis bezüglich ψ in ein Ergebnis bezüglich φ zu überführen.

ein kompositionaler $\langle \psi, r, \varphi \rangle$ -Repräsentant g.d.w. für alle $R \in \mathcal{U}$ gilt: $L \oplus U$ ist ein $\langle \psi, r, \varphi \rangle$ -Repräsentant. Ist \sim_φ eine Kongruenz auf $\langle \mathcal{U}, \oplus \rangle$ (d.h. \oplus und \sim_φ sind verträglich), dann können wir kompositionale Repräsentanten wie folgt einsetzen, um ein beliebiges $M \oplus R \in \mathcal{U}$ zu analysieren:

- (1) Überführe M in einen kompositionalen $\langle \psi, r, \varphi \rangle$ -Repräsentanten L mit $L \sim_\varphi M$.
- (2) Dann ist $L \oplus R$ ein $\langle \psi, r, \varphi \rangle$ -Repräsentant und aus der Kongruenz folgt: $L \oplus R \sim_\varphi M \oplus R$.
- (3) Wir lösen das Problem φ für $L \oplus R$, wie weiter oben beschrieben.

2 Mathematische Grundlagen

2.1 Natürliche und reelle Zahlen

Wir notieren die Menge der natürlichen Zahlen $0, 1, 2, \dots$ als \mathbb{N}_0 und schreiben \mathbb{N}_1 für $\mathbb{N}_0 \setminus \{0\}$. Wir schreiben \mathbb{R} für die Menge der reellen Zahlen. Die Symbole $-\infty$ und ∞ haben ihre übliche Bedeutung: $-\infty < x < \infty$ für alle $x \in \mathbb{R}$.

2.2 Wörter und Sprachen

Sei Σ eine Menge und $n \in \mathbb{N}_0$. Dann schreiben wir Σ^n für die Menge der *Wörter* (auch: Folgen, Sequenzen) $w = a_1 \dots a_n$ mit $a_i \in \Sigma$ für alle $1 \leq i \leq n$. Wir schreiben $|w|$ für die Länge n des Wortes w . Wir schreiben ε für das leere Wort mit $\Sigma^0 = \{\varepsilon\}$ und $|\varepsilon| = 0$. Wie üblich definieren wir:

$$\begin{aligned}\Sigma^+ &:= \bigcup_{n \in \mathbb{N}_1} \Sigma^n \\ \Sigma^* &:= \Sigma^0 \cup \Sigma^+ .\end{aligned}$$

2.3 Funktionen, Familien und Multimengen

Wir schreiben B^A für die Klasse aller Abbildungen $f: A \rightarrow B$ von der Menge A in die Menge B . Wir schreiben $\text{DEF}(f)$ für den *Definitionsbereich* von f . Eine Abbildung $f: A \rightarrow B$ ist *injektiv*, falls für alle $b \in B$ höchstens ein $a \in A$ mit $f(a) = b$ existiert. Eine Abbildung $f: A \rightarrow B$ heißt *surjektiv*, falls für alle $b \in B$ wenigstens ein $a \in A$ mit $f(a) = b$ existiert. Eine injektive und surjektive Abbildung f ist *bijektiv*. Wenn möglich, fassen wir $f: A \rightarrow B$ auch als die Menge $\{\langle a, f(a) \rangle \mid a \in A\}$, auch geschrieben als $\{a \mapsto f(a) \mid a \in A\}$, auf. Sind f und g Funktionen mit disjunkten Domänen, bildet ihre Vereinigung $f \cup g$ eine neue Funktion.

Sind $f: A \rightarrow B$ und $g: B \rightarrow C$ Funktionen, schreiben wir $f \circ g$ für die Komposition von f und g , also die Funktion $A \rightarrow C$ mit $a \mapsto g(f(a))$ für alle $a \in A$.

Sei $f: A \rightarrow B$ und $A \subseteq A'$. Dann ist f eine *partielle Abbildung* von der Menge A' in die Menge B , geschrieben $f: A' \rightsquigarrow B$. Wir nennen dann f *definiert* für alle $a \in \text{DEF}(f) = A$.

2.3.1 Familien

Seien I und \mathbb{K} Mengen. Sei $f: I \rightarrow \mathbb{K}$ eine Funktion. Dann nennen wir f auch *Familie* über *Indexmenge* I und *Wertebereich* \mathbb{K} . Wir schreiben $\text{FAM}_I(\mathbb{K})$ bzw. $\text{FAM}(\mathbb{K})$ für die Klasse aller Familien über Indexmenge I und Wertebereich \mathbb{K} bzw. aller Familien über beliebigen Indexmengen und Wertebereich \mathbb{K} . Eine Familie heißt *endlich*, falls ihre Indexmenge endlich ist. Eine Familie heißt *konstant*, falls sie für jeden Index den gleichen Wert annimmt, also für alle Indizes $i, i' \in I$ gilt: $f(i) = f(i') = a \in \mathbb{K}$. Zu jedem $a \in \mathbb{K}$ notieren wir die entsprechende konstante Familie mit \mathbf{a}_I . Ist I eine einelementige Menge, schreiben wir auch \mathbf{a} anstatt \mathbf{a}_I .

Eine Familie heißt *beschränkt*, wenn sie nur endlich viele unterschiedliche Werte annimmt, also die Menge $\{f(i) \mid i \in I\}$ endlich ist. Eine Familie heißt *leer*, falls ihre Indexmenge leer ist. Wir identifizieren alle leeren Familien und notieren *die* leere Familie als \emptyset .

2.3.1.1 Operationen und Relationen auf Familien

Sei $*$: $\mathbb{K} \times \mathbb{K} \rightsquigarrow \mathbb{K}$ eine partielle zweistellige Abbildung. Dann übertragen wir $*$ kanonisch auf Familien über beliebigen Indexmengen und Wertebereich \mathbb{K} . Seien $f \in \text{FAM}_I(\mathbb{K})$ und $g \in \text{FAM}_J(\mathbb{K})$, sodass für alle $x \in I \cap J$ gilt: $\langle f(x), g(x) \rangle \in \text{DEF}(*)$. Dann definieren wir $f * g \in \text{FAM}_{I \cup J}(\mathbb{K})$ durch

$$(f * g)(x) := \begin{cases} f(x) * g(x), & \text{falls } x \in I \cap J \\ f(x), & \text{falls } x \in I \setminus J \\ g(x), & \text{sonst} \end{cases} \quad (1)$$

Sei $a \in \mathbb{K}$. Falls $\langle a, f(i) \rangle \in \text{DEF}(*)$ für alle $i \in I$ gilt, definieren wir

$$a * f := \mathbf{a}_I * f \quad (2)$$

$$f * a := f * \mathbf{a}_I \quad (3)$$

Außerdem definieren wir die a -Erweiterung $f|_J^a \in \text{FAM}_J(\mathbb{K})$ von f auf J durch

$$f(j) := \begin{cases} f(j), & \text{falls } j \in I \\ a, & \text{sonst} \end{cases} \quad (4)$$

Sei weiterhin $\leq \subseteq \mathbb{K} \times \mathbb{K}$. Dann definieren wir

$$f \leq_a g \text{ g.d.w. } f|_{I \cup J}^a(x) \leq g|_{I \cup J}^a(x) \text{ für alle } x \in I \cup J \quad (5)$$

2.3.1.2 Familien-Isomorphismen

Seien \mathbb{K} und \mathbb{K}' Mengen und $f \in \text{FAM}_I(\mathbb{K})$ und $g \in \text{FAM}_J(\mathbb{K}')$ Familien. Seien $h: I \rightarrow J$ und $k: \mathbb{K} \rightarrow \mathbb{K}'$ Bijektionen, sodass $k(f(i)) = g(h(i))$. Dann ist $\langle h, k \rangle$ ein *Familien-Isomorphismus* von f nach g .

Falls ein Familien-Isomorphismus $\langle h, k \rangle$ von f nach g existiert, nennen wir f und g *isomorph*, notiert als $f \sim_{h,k} g$ oder kurz $f \sim g$.

Ein Familienisomorphismus $\langle h, k \rangle$ heißt *Index-Isomorphismus*, falls k die Identität ist, also $k(a) = a$ für alle $a \in \mathbb{K}$ gilt. Zwei Familien heißen *index-isomorph*, falls ein Index-Isomorphismus von f nach g existiert.

2.3.2 Kompressoren

In dieser Arbeit betrachten wir partielle Funktionen, die eine Familie von Werten über einer Domäne zu einem einzelnen Wert über der Domäne zusammenfassen, also eine Familie in einen Wert *komprimieren*.

Eine partielle Funktion $\kappa: \text{FAM}(\mathbb{K}) \rightsquigarrow \mathbb{K}$ nennen wir einen *Kompressor* über \mathbb{K} , falls jede der folgenden Eigenschaften erfüllt ist:

- (1) *Index-Unabhängigkeit*: Für zwei index-isomorphe Familien f und g gilt: κ ist für f durch a definiert g.d.w. κ für g durch a definiert ist.
- (2) *Identitäts-Erhaltung*: κ ist für jedes $a \in \mathbb{K}$ und jede konstant Familie \mathbf{a} über einer einelementigen Indexmenge als a definiert.

Wir definieren im Folgenden einige Eigenschaften von Kompressoren.

Sei κ ein Kompressor, sodass für jede Familie $f \in \text{FAM}_I(\mathbb{K})$ und jede Zerlegung $g \in \text{FAM}_J(\text{FAM}(\mathbb{K}))$ von f in Teilfamilien gilt:

- (1) $f \in \text{DEF}(\kappa)$ g.d.w. $g(j) \in \text{DEF}(\kappa)$ für alle $j \in J$,
- (2) Falls $f \in \text{DEF}(\kappa)$, gilt: $\kappa(f) = \kappa((a_j)_{j \in J})$ mit $a_j = \kappa(g(j))$.

Dann heißt κ *kompositional*.

Ein Kompressor κ heißt *semitotal*, falls κ für alle endlichen Familien definiert ist.

Sei ω ein Symbol. Dann ist die ω -*Totalisierung* von κ auf \mathbb{K} der totale Kompressor über $\mathbb{K} \cup \{\omega\}$ mit

$$\kappa(f) = \begin{cases} a, & \text{falls } f \in \text{FAM}(\mathbb{K}) \text{ und } \kappa \text{ für } f \text{ durch } a \text{ definiert ist} \\ \omega, & \text{sonst} \end{cases} \quad (6)$$

2.3.3 Multimengen

Eine Familie $m \in \text{FAM}_I(\mathbb{N}_0)$ ist eine *Multimenge* über I . Wir notieren die Klasse aller Multimengen über I als $\text{BAGS}(I)$.

Seien $m \in \text{BAGS}(I)$ und $n \in \text{BAGS}(J)$ Multimengen. Wir schreiben auch $m \sqsubset n$ bzw. $m \sqsubseteq n$, falls $m <_0 n$ bzw. $m \leq_0 n$. Zwei Multimengen m und n heißen *äquivalent*, falls $m \sqsubseteq n$ und $m \sqsupseteq n$.

Der *Träger* $\text{SUPP}(m)$ einer Multimenge $m \in \text{BAGS}(I)$ ist die Menge aller Indizes mit Wert größer 0, das heißt,

$$\text{SUPP}(m) := \{i \in I \mid m(i) > 0\} . \quad (7)$$

Eine Multimenge $m \in \text{BAGS}(I)$ heißt *minimal*, falls $\text{SUPP}(m) = I$. Zu jeder Multimenge $m \in \text{BAGS}(I)$ existiert offenbar eine minimale Multimenge $m' \in \text{BAGS}(\text{SUPP}(m))$ mit $m'(x) = m(x)$ für alle $x \in \text{SUPP}(m)$. Der Einfachheit halber identifizieren wir im folgenden eine Multimenge stets mit ihrer minimalen Multimenge. Folglich identifizieren wir alle äquivalenten Multimengen. Insbesondere identifizieren wir auch alle leeren Multimengen und notieren die leere Multimenge als \emptyset .

Wir beobachten, dass $m + n$ und $m \cdot n$ für beliebige Multimengen m und n definiert ist. Die Multimengen-Addition ist außerdem assoziativ und kommutativ. Dagegen ist $m - n$ genau für Multimengen m und n mit $m \sqsupseteq n$ definiert. Weiterhin ist \sqsubseteq eine Halbordnung mit kleinstem Element \emptyset . Weiterhin gelten für Multimengen m, n, p mit $m \sqsupseteq n$: $m - \emptyset = m$, $m \cdot \emptyset = \emptyset$, $(m - n) + n = m$ und $p \cdot (m + n) = (p \cdot m) + (p \cdot n)$.

2.4 Ordnungstheoretische Grundlagen

2.4.1 Quasiordnungen

Sei \mathbb{K} eine Menge und $\leq, <$ binäre Relationen auf \mathbb{K} .

Dann ist \leq eine *Quasiordnung* auf \mathbb{K} , falls

- (1) \leq *reflexiv* ist, das heißt, für alle $a \in \mathbb{K}$ gilt $a \leq a$, und
- (2) \leq *transitiv* ist, das heißt, für alle $a, b, c \in \mathbb{K}$ mit $a \leq b$ und $b \leq c$ gilt $a \leq c$.

Dagegen ist $<$ eine *strikte Quasiordnung* auf \mathbb{K} , falls

- (1) $<$ *irreflexiv* ist, das heißt, für alle $a \in \mathbb{K}$ gilt $a \not< a$, und
- (2) $<$ *transitiv* ist.

2.4.2 Halbgeordnete Mengen

Sei \mathbb{K} eine Menge und \leq eine Quasiordnung auf \mathbb{K} . Dann ist \leq eine *Halbordnung* auf \mathbb{K} und $\langle \mathbb{K}, \leq \rangle$ eine *halbgeordnete Menge*, falls \leq *antisymmetrisch* ist, das heißt, für alle $a, b \in \mathbb{K}$ mit $a \leq b$ und $b \leq a$ gilt $a = b$.

Gleichermaßen ist eine strikte Quasiordnung $<$ auf \mathbb{K} eine strikte Halbordnung, falls $<$ antisymmetrisch ist. Wir beobachten, dass es zu jeder Halbordnung genau eine strikte Halbordnung gibt und vice versa, so dass $a < b$ g.d.w. $a \leq b$ und $a \neq b$.

Notation 2 (Halbordnung/Strikte Halbordnung)

Sei \leq eine Halbordnung auf \mathbb{K} . Falls klar aus dem Kontext schreiben wir $<$ für die strikte Halbordnung auf \mathbb{K} mit $a < b$ g.d.w. $a \leq b$ und $a \neq b$.

Sei $<$ eine strikte Halbordnung auf \mathbb{K} . Falls klar aus dem Kontext schreiben wir \leq für die Halbordnung mit $a \leq b$ g.d.w. $a < b$ oder $a = b$.

Eine Halbordnung \leq auf \mathbb{K} ist eine *totale Ordnung* auf \mathbb{K} , falls $a \leq b$ oder $b \leq a$ für alle $a, b \in \mathbb{K}$. Eine strikte Halbordnung $<$ auf \mathbb{K} ist eine *Striktordnung* auf \mathbb{K} , falls \leq eine totale Ordnung ist.

2.4.3 Verbände

Sei $\langle \mathbb{K}, \leq \rangle$ eine halbgeordnete Menge. Ein Element $k \in \mathbb{K}$ ist eine *obere Schranke* einer Menge $A \subseteq \mathbb{K}$ falls für alle $a \in A$ gilt: $a \leq k$. Eine obere Schranke ist ein *Supremum* einer Menge $A \subseteq \mathbb{K}$, falls für alle oberen Schranken k' von A gilt: $k \leq k'$. Es gibt stets höchstens ein Supremum für eine Menge $A \subseteq \mathbb{K}$. Ist das Supremum k von A in A enthalten, ist k das *Maximum* von A . Analog dazu verwenden wir die Begriffe der unteren Schranke, des Infimums und des Minimums.

$\langle \mathbb{K}, \leq \rangle$ ist ein *Verband*, falls jede nicht-leere, endliche Teilmenge ein Supremum und ein Infimum besitzt. Ein Verband ist ein *vollständiger Verband*, falls jede Teilmenge ein Supremum und ein Infimum besitzt. Dabei ist die Existenz aller Suprema hinreichend für die Existenz aller Infima. Außerdem ist ein vollständiger Verband stets *beschränkt*, das heißt, die gesamte Menge hat ein Minimum und ein Maximum.

Wir verwenden die Begriffe obere/untere Schranke, Supremum/Infimum, Maximum/Minimum auch für Familien, indem wir die Familie als Menge auffassen.

Sei für $i = 1, 2$ die Struktur $\langle \mathbb{K}_i, \leq_i \rangle$ ein Verband. Wir definieren das *Kreuzprodukt* $\langle \mathbb{K}_1, \leq_1 \rangle \times \langle \mathbb{K}_2, \leq_2 \rangle := \langle \mathbb{K}, \leq \rangle$ von $\langle \mathbb{K}_1, \leq_1 \rangle$ und $\langle \mathbb{K}_2, \leq_2 \rangle$ durch

$$\mathbb{K} := \mathbb{K}_1 \times \mathbb{K}_2, \quad (8)$$

$$\leq := \{ \langle \langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle \rangle \mid x_i, y_i \in \mathbb{K}_i, i = 1, 2, x_i \leq_i y_i \}. \quad (9)$$

Wir wiederholen den Beweis für das bekannte Lemma, dass das Kreuzprodukt zweier (vollständiger) Verbände erneut ein (vollständiger) Verband ist.

Lemma 3

Sei für $i = 1, 2$ die Struktur $\langle \mathbb{K}_i, \leq_i \rangle$ ein Verband.

Für $\langle \mathbb{K}, \leq \rangle = \langle \mathbb{K}_1, \leq_1 \rangle \times \langle \mathbb{K}_2, \leq_2 \rangle$ gilt:

- (1) $\langle \mathbb{K}, \leq \rangle$ ist ein Verband.
- (2) Seien $\langle \mathbb{K}_1, \leq_1 \rangle, \langle \mathbb{K}_2, \leq_2 \rangle$ vollständig. Dann ist $\langle \mathbb{K}, \leq \rangle$ vollständig.

Beweis (Lemma 3)

Sei $\langle \mathbb{K}_1, \leq_1 \rangle \times \langle \mathbb{K}_2, \leq_2 \rangle = \langle \mathbb{K}, \leq \rangle$.

- (1) a) Wir zeigen, dass \leq eine Halbordnung ist.
 - i. \leq ist reflexiv. Da \leq_i eine Halbordnung ist, ist \leq_i reflexiv. Es gilt also $x_i \leq_i x_i$, also auch $\langle x_1, x_2 \rangle \leq \langle x_1, x_2 \rangle$.
 - ii. \leq ist transitiv. Da \leq_i eine Halbordnung ist, ist \leq_i transitiv. Seien x_i, y_i, z_i so gewählt, dass gilt: $\langle x_1, x_2 \rangle \leq \langle y_1, y_2 \rangle$ und $\langle y_1, y_2 \rangle \leq \langle z_1, z_2 \rangle$. Also gelten auch $x_i \leq_i z_i$ und damit auch $\langle x_1, x_2 \rangle \leq \langle z_1, z_2 \rangle$.
 - iii. \leq ist antisymmetrisch. Da \leq_i eine Halbordnung ist, ist \leq_i antisymmetrisch. Also gilt $x_i \leq_i y_i$ impliziert $x_i = y_i$ oder $y_i \not\leq_i x_i$. Seien x_i, y_i so gewählt, dass gilt: $\langle x_1, x_2 \rangle \leq \langle y_1, y_2 \rangle$. Also gilt $x_i \leq_i y_i$. Also gilt $x_i = y_i$ oder $y_i \not\leq_i x_i$. Also gilt $\langle y_1, y_2 \rangle \not\leq \langle x_1, x_2 \rangle$ oder $\langle y_1, y_2 \rangle = \langle x_1, x_2 \rangle$.
- b) Wir zeigen, dass für zwei beliebige Elemente $\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle$ ein Supremum existiert. Da \leq_i ein Verband ist, existiert ein Supremum z_i von x_i, y_i . Wir zeigen, dass $\langle z_1, z_2 \rangle$ ein Supremum von $\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle$ ist. Da z_i ein Supremum von x_i, y_i ist, gilt $x_i, y_i \leq_i z_i$. Also gilt $\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle \leq \langle z_1, z_2 \rangle$. Also ist $\langle z_1, z_2 \rangle$ eine obere Schranke. Ist $\langle w_1, w_2 \rangle$ eine obere Schranke von $\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle$ gilt $x_i, y_i \leq_i w_i$. Also ist w_i eine obere Schranke von x_i, y_i . Da z_i ein Supremum ist, gilt $z_i \leq_i w_i$. Also gilt $\langle z_1, z_2 \rangle \leq \langle w_1, w_2 \rangle$. Also ist $\langle z_1, z_2 \rangle$ ein Supremum von $\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle$.

- c) Für Infimum geht der Beweis analog.
- (2) Wir zeigen, dass auch die Vollständigkeit erhalten bleibt. Sei $X_1 \times X_2$ mit $X_i \subseteq \mathbb{K}_i$. Sei z_i das Supremum der Menge X_i . Wir zeigen, dass $\langle z_1, z_2 \rangle$ das Supremum der Menge $X_1 \times X_2$ ist.
- a) Wir zeigen, dass $\langle z_1, z_2 \rangle$ eine obere Schranke der Menge $X_1 \times X_2$ ist. Da z_i ein Supremum von X_i ist, ist z_i eine obere Schranke von X_i . Also gilt für alle $x_i \in X_i$: $x_i \leq z_i$. Also gilt für alle $\langle x_1, x_2 \rangle \in X_1 \times X_2$: $\langle x_1, x_2 \rangle \leq \langle z_1, z_2 \rangle$.
- b) Wir zeigen, dass für alle oberen Schranken $\langle w_1, w_2 \rangle$ von $X_1 \times X_2$ gilt: $\langle z_1, z_2 \rangle \leq \langle w_1, w_2 \rangle$. Da $\langle w_1, w_2 \rangle$ eine obere Schranke von $X_1 \times X_2$ ist, gilt $\langle x_1, x_2 \rangle \leq \langle w_1, w_2 \rangle$ für alle $\langle x_1, x_2 \rangle \in X_1 \times X_2$. Also gilt $x_i \leq w_i$ für alle $x_i \in X_i$. Also ist w_i eine obere Schranke von X_i . Da z_i ein Supremum ist, gilt $z_i \leq w_i$. Also gilt $\langle w_1, w_2 \rangle \leq \langle z_1, z_2 \rangle$.

2.5 Partiiell vollständige Strukturen

Wie üblich betrachten wir *Strukturen* (auch: Algebren) als Objekte, die aus Mengen sowie Operationen und Relationen auf diesen Mengen bestehen.

In diesem Abschnitt definieren wir *partiell vollständige Strukturen* [Wei88]. Intuitiv heißt eine Struktur *vollständig*, wenn die Operationen auch für unendliche Familien von Eingaben definiert sind. Eine vollständige Struktur sind die natürlichen Zahlen mit der Minimum-Operation: Selbst unendliche Familien natürlicher Zahlen haben ein Minimum. Im Gegensatz dazu ist die klassische Addition auf unendlichen Familien natürlicher Zahlen nur partiell, daher ist diese Struktur nur *partiell vollständig*.

Wir beschränken uns auf die Definition zweier Klassen partiell vollständiger Strukturen: *Partiiell vollständige kommutative Monoide* und die darauf basierenden *partiell vollständigen Bewertungs-Monoide* und *partiell vollständigen Multi-Hemiringe*. Dazu wiederholen wir zunächst die zugrunde liegenden algebraischen Strukturen der kommutativen Monoide, sowie der von Droste und Meinecke [DM11] definierten Bewertungs-Monoide und die von Droste und Kuich [DK13] definierten Multi-Hemiringe.

2.5.1 Kommutative Monoide, Bewertungs-Monoide und Multi-Hemiringe

Wir definieren im Folgenden die aus der Literatur bekannten Begriffe der *Monoide*, der *Bewertungs-Monoide* und der *Multi-Hemiringe*.

Definition 4 (Monoid)

Ein *Monoid* $\langle \mathbb{K}, +, 0 \rangle$ besteht aus einer Trägermenge \mathbb{K} , einer assoziativen binären Operation $+$ auf \mathbb{K} sowie einem Element 0 mit $a + 0 = 0 + a = a$ für alle $a \in \mathbb{K}$. Ein Monoid heißt *kommutativ*, falls $+$ kommutativ ist.

Notation 5 (Anwendung eines Monoids auf Folgen)

Sei $\langle \mathbb{K}, +, 0 \rangle$ ein Monoid. Wir notieren die Abbildung $\mathbb{K}^+ \rightarrow \mathbb{K}$ mit

$$a_1 \dots a_n \mapsto a_1 + \dots + a_n$$

als $+$.

Ein Bewertungsmonoid ist ein kommutatives Monoid zusammen mit einer Bewertungsfunktion:

Definition 6 (Bewertungs-Monoid)

Ein *Bewertungs-Monoid* $\langle \mathbb{K}, +, \text{val}, 0 \rangle$ besteht aus einem kommutativen Monoid $\langle \mathbb{K}, +, 0 \rangle$ und einer Funktion $\text{val}: \mathbb{K}^+ \rightarrow \mathbb{K}$ mit $\text{val}(a) = a$ für alle $a \in \mathbb{K}$ und $\text{val}(w) = 0$ für alle $w = a_1 \dots a_n \in \mathbb{K}^+$ mit wenigstens einem $a_i = 0$ für $1 \leq i \leq n$.

Ein Multi-Hemiring ist ein kommutatives Monoid zusammen mit einer Familie von Produkten über der Indexmenge $\mathbb{N}_1 \times \mathbb{N}_1$.

Definition 7 (Multi-Hemiring)

Ein *Multi-Hemiring* $\langle \mathbb{K}, +, \circ, 0 \rangle$ besteht aus einem kommutativen Monoid $\langle \mathbb{K}, +, 0 \rangle$ und einer Familie

$$\circ = (\cdot_{m,n})_{m,n \geq 1}$$

binärer Operationen $\cdot_{m,n}$ auf \mathbb{K} , so dass die folgenden Gleichungen für alle $m, n, p \geq 1$ und $a, b, c \in \mathbb{K}$ gelten:

$$(a \cdot_{m,n} b) \cdot_{m+n,p} c = a \cdot_{m,n+p} (b \cdot_{n,p} c) \quad (10)$$

$$a \cdot_{m,n} 0 = 0 \cdot_{m,n} a = 0 \quad (11)$$

$$a \cdot_{m,n} (b + c) = (a \cdot_{m,n} b) + (a \cdot_{m,n} c) \quad (12)$$

$$(a + b) \cdot_{m,n} c = (a \cdot_{m,n} c) + (b \cdot_{m,n} c) \quad (13)$$

Ein Multi-Hemiring ist ein *Hemiring*, falls für alle $m, n, p, q \geq 1$ und $a, b \in \mathbb{K}$ gilt:

$$a \cdot_{m,n} b = a \cdot_{p,q} b \quad (14)$$

In einem Hemiring schreiben wir daher auch $a \cdot b$ anstatt $a \cdot_{m,n} b$.

Ein *Semiring* $\langle H, \mathbb{1} \rangle$ besteht aus einem Hemiring H und einem Element $\mathbb{1} \in \mathbb{K}$, sodass $a \cdot \mathbb{1} = \mathbb{1} \cdot a = a$ für alle $a \in \mathbb{K}$.

Außerdem sind Multi-Hemiringe spezielle Bewertungs-Monoide. Dazu führen wir zunächst die folgende Notation ein, um die Produkte eines Multi-Hemirings rekursiv auf eine Folge von Werten anzuwenden:

Notation 8 (Anwendung eines Multi-Hemirings auf Folgen)

Sei $\langle \mathbb{K}, +, \circ, \mathbb{0} \rangle$ ein Multi-Hemiring. Sei $\alpha \in \mathbb{K}$. Wir notieren die Abbildung $f: \mathbb{K}^+ \rightarrow \mathbb{K}$ mit

$$a \mapsto a \text{ für alle } a \in \mathbb{K} \quad (15)$$

$$aw \mapsto a \cdot_{1, |w|} f(w) \text{ für alle } w \in \mathbb{K}^+, a \in \mathbb{K} \quad (16)$$

als \circ^+ .

Wenden wir diese Notation an, induziert ein Multi-Hemiring $\langle \mathbb{K}, +, \circ, \mathbb{0} \rangle$ das Bewertungs-Monoid $\langle \mathbb{K}, +, \circ^+, \mathbb{0} \rangle$.

2.5.2 Partiiell vollständige kommutative Monoide und Multi-Hemiringe

Wir betrachten die algebraische Struktur des *partiell vollständigen kommutativen Monoid*. Es handelt sich hierbei um eine Verallgemeinerung eines kommutativen Monoids um partielle Summierbarkeit, das heißt, einigen unendlichen Summen sind Werte zugeordnet.

Definition 9 (Partiiell vollständiges Monoid)

Sei \mathbb{K} eine Menge. Sei \sum ein semitotaler, kompositionaler Kompressor über \mathbb{K} . Sei $\mathbb{0} \in \mathbb{K}$ und sei \sum für die leere Familie \emptyset als $\mathbb{0}$ definiert.

Dann nennen wir $\langle \mathbb{K}, \sum, \mathbb{0} \rangle$ ein *partiell vollständiges kommutatives Monoid*. \sum heißt *Addition*, $\mathbb{0}$ heißt *neutrales Element* von $\langle \mathbb{K}, \sum, \mathbb{0} \rangle$.

Ist \sum ein totaler Kompressor, nennen wir $\langle \mathbb{K}, \sum, \mathbb{0} \rangle$ *vollständiges kommutatives Monoid*.

Beispiel 10

Die folgenden Strukturen sind partiell vollständige kommutative Monoide:

- (1) $N_{\max} := \langle \mathbb{N}_0 \cup \{-\infty\}, \max, -\infty \rangle$
- (2) $N_{\sup} := \langle \mathbb{N}_0 \cup \{-\infty, +\infty\}, \sup, -\infty \rangle$
- (3) $N_{\min} := \langle \mathbb{N}_0 \cup \{+\infty\}, \min, +\infty \rangle$
- (4) $N_{\inf} := \langle \mathbb{N}_0 \cup \{+\infty\}, \inf, +\infty \rangle$

Die Operation \max auf $\mathbb{N} \cup \{-\infty\}$ ist wie üblich nur für endliche Indexmengen definiert. Die Operation \sup auf $\mathbb{N} \cup \{-\infty, +\infty\}$ dagegen ist für alle Indexmengen total definiert: Die Anwendung von \sup auf eine unendliche Menge hat als Ergebnis $+\infty$. Die Operation \min auf $\mathbb{N} \cup \{+\infty\}$ ist dagegen für alle unendlichen Mengen definiert, da die Menge \mathbb{N} wohlgeordnet ist und es daher keine unendlichen absteigenden Ketten gibt. Daher fallen die Strukturen N_{\min} und N_{\inf} zusammen.

Wie üblich schreiben wir anstelle von $\sum((a_i)_{i \in I})$ auch

$$\sum_{i \in I} a_i . \quad (17)$$

Jedes partiell vollständige kommutative Monoid $\langle \mathbb{K}, \sum, 0 \rangle$ induziert das kommutative Monoid $\langle \mathbb{K}, +, 0 \rangle$ mit $a_i + a_j := \sum_{n \in \{i, j\}} a_n$. Kommutativität und Assoziativität folgen direkt aus der Index-Unabhängigkeit und der Kompositionalität von \sum .

Anders herum können wir zu jedem kommutativen Monoid $\langle \mathbb{K}, \sum, 0 \rangle$ kanonisch ein partiell vollständiges Monoid bilden, seine *minimale Generalisierung*, indem wir \sum für alle Familien $(a_i)_{i \in I}$ über endlichen Indexmengen $I = \{i_1, \dots, i_n\}$ als $a_{i_1} + \dots + a_{i_n}$ definieren.

Aufbauend auf der minimalen Generalisierung definieren wir die *kanonische Generalisierung*, die noch diejenigen Familien hinzufügt, die die Wiederholung idempotenter Elemente ignorieren: Sei A die Menge aller $a \in \mathbb{K}$ mit $a + a = a$. In der kanonischen Generalisierung sei \sum für alle beschränkten Familien $f = (a_i)_{i \in I}$ über Indexmenge I definiert, für die gilt: Für jedes $a \in A$ existieren nur endlich viele i mit $a_i = a$. Sei $J \subseteq I$, sodass für alle $a \in \mathbb{K}$ gilt:

- (1) Ist $a \in A$, ist die Anzahl der $i \in I$ mit $a_i = a$ gleich der Anzahl der $j \in J$ mit $a_j = a$.
- (2) Ist $a \notin A$ und existiert wenigstens ein $i \in I$ mit $a_i = a$, dann existiert genau ein $j \in J$ mit $a_j = a$.

Dann ist J endlich. Wir definieren dann den Wert von $\sum(f)$ wie die Summe von $(a_j)_{j \in J}$ in der minimalen Generalisierung.

Basierend auf partiell vollständigen kommutativen Monoiden generalisieren wir Bewertungs-Monoiden (Multi-Hemiringe) zu *partiell vollständigen Bewertungs-Monoiden* (Multi-Hemiringen).

Definition 11 (Partiell vollständiges Bewertungs-Monoid)

Sei $\langle \mathbb{K}, \sum, 0 \rangle$ ein partiell vollständiges, kommutatives Monoid mit induziertem Monoid $\langle \mathbb{K}, +, 0 \rangle$. Sei $\langle \mathbb{K}, +, \text{val}, 0 \rangle$ ein Bewertungs-Monoid.

Dann ist $\langle \mathbb{K}, \sum, \text{val}, 0 \rangle$ ein *partiell vollständiges Bewertungs-Monoid*.

Ein partiell vollständiges Bewertungs-Monoid $\langle \mathbb{K}, \sum, \text{val}, 0 \rangle$ induziert einen Multi-Hemiring $\langle \mathbb{K}, +, \text{val}, 0 \rangle$, indem wir $+$ genau wie im von $\langle \mathbb{K}, \sum, 0 \rangle$ induzierten kommutativen Monoid definieren. Ebenfalls ist es möglich aus einem Bewertungs-Monoid kanonisch ein partiell vollständiges Bewertungs-Monoid zu bilden, indem genauso verfahren wird wie bei der Generalisierung eines kommutativen Monoids.

Definition 12 (Partiell vollständiger Multi-Hemiring)

Sei $\langle \mathbb{K}, \sum, 0 \rangle$ ein partiell vollständiges Monoid. Sei $\circ = (\cdot_{m,n})_{m,n \geq 1}$ eine Familie von Funktionen $\cdot_{m,n} : \mathbb{K} \times \mathbb{K} \rightarrow \mathbb{K}$, sodass die folgenden Bedingungen gelten:

- (1) $(a \cdot_{m,n} b) \cdot_{m+n,p} c = a \cdot_{m,n+p} (b \cdot_{n,p} c)$.
- (2) $a \cdot_{m,n} 0 = 0 \cdot_{m,n} a = 0$.
- (3) Sei $f = (a_i)_{i \in I} \in \text{FAM}(\mathbb{K})$. Dann gilt:
 - a) $\sum_{i \in I} a_i$ ist definiert g.d.w. für alle $b \in \mathbb{K}$ und $m, n \geq 1$ jeweils $\sum_{i \in I} (a_i \cdot_{m,n} b)$ und $\sum_{i \in I} (b \cdot_{m,n} a_i)$ definiert sind.
 - b) Falls $\sum_{i \in I} a_i$ definiert ist, gilt für alle $b \in \mathbb{K}$ und $m, n \geq 1$ jeweils $b \cdot_{m,n} \sum_{i \in I} a_i = \sum_{i \in I} (b \cdot_{m,n} a_i)$ und $(\sum_{i \in I} a_i) \cdot_{m,n} b = \sum_{i \in I} (a_i \cdot_{m,n} b)$.

Dann nennen wir $\langle \mathbb{K}, \sum, \circ, 0 \rangle$ einen *partiell vollständigen Multi-Hemiring*.

Ein partiell vollständiger Multi-Hemiring $\langle \mathbb{K}, \sum, \circ, 0 \rangle$ induziert einen Multi-Hemiring $\langle \mathbb{K}, +, \circ, 0 \rangle$, indem wir $+$ genau wie im von $\langle \mathbb{K}, \sum, 0 \rangle$ induzierten kommutativen Monoid definieren. Ebenfalls ist es möglich aus einem Multi-Hemiring kanonisch einen partiell vollständigen Multi-Hemiring zu bilden, indem genauso verfahren wird wie bei der Generalisierung eines kommutativen Monoids.

2.6 Automaten

Ein *Automat* $A = \langle Q, I, F, \Sigma, \cdot \rightarrow \cdot \rangle$ über einem *Alphabet* Σ besteht aus einer Menge Q von *Zuständen*, einer Menge $I \subseteq Q$ von *Anfangszuständen*, einer Menge $F \subseteq Q$ von *Endzuständen* und einer *Transitionsrelation* $\cdot \rightarrow \cdot \subseteq Q \times \Sigma \times Q$.

Auf Basis von $\cdot \rightarrow \cdot$ definieren wir $\cdot \Rightarrow \cdot \subseteq Q \times \Sigma^* \times Q$ rekursiv durch

$$p \xRightarrow{\varepsilon} p, \text{ falls } p \in Q \quad (18)$$

$$p \xRightarrow{wa} r, \text{ falls ein } q \in Q \text{ existiert mit } p \xRightarrow{w} q \text{ und } q \xrightarrow{a} r. \quad (19)$$

Ein Automat A *akzeptiert* ein Wort $w \in \Sigma^*$, falls ein $p \in I$ und ein $q \in F$ existieren mit $p \xRightarrow{w} q$.

Wir schreiben $p \xrightarrow{a}$ bzw. $p \xRightarrow{w}$, falls ein q existiert mit $p \xrightarrow{a} q$ bzw. $p \xRightarrow{w} q$. Wir schreiben $p \rightarrow$, falls ein a existiert mit $p \xrightarrow{a}$. Analog führen wir die Schreibweisen $\xrightarrow{a} q$, $\rightarrow q$ und $\xRightarrow{w} q$ ein.

Ein Automat heißt *abzählbar* bzw. *endlich*, falls $Q \cup \Sigma$ abzählbar bzw. endlich ist. Ein Automat A ist *deterministisch*, falls (1) $|I| = 1$ und (2) seine Transitionsrelation als Funktion $Q \times \Sigma \rightarrow Q$ aufgefasst werden kann.

2.7 Petrinetze

Wir setzen eine Grundkenntnis über den Formalismus der *Petrinetze* zur Modellierung von Verhalten voraus, für eine detaillierte Einführung in den Formalismus empfehlen wir zum Beispiel [Rei13]. In diesem Abschnitt geben wir die in dieser Arbeit benötigten, aus der Literatur wohlbekannten Begriffe und Notationen wieder.

2.7.1 Netzstrukturen und Petrinetze

Für ausführliche Definitionen und Beispiele verweisen wir auf [Rei13].

2.7.1.1 Netzstrukturen

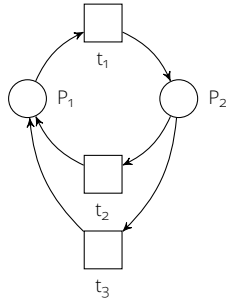
Eine *Netzstruktur* besteht aus *Plätzen* und *Transitionen*, die so durch gewichtete Kanten miteinander verbunden sind, dass ein bipartiter Graph entsteht. Ein Platz modelliert einen Speicher oder eine Bedingung, eine Transition eine Aktion oder ein Ereignis. Eine *Markierung* weist jedem Platz eine nicht-negative Anzahl von *Marken* zu.

Definition 13 (Netzstruktur, Markierung)

Eine *Netzstruktur* $S = \langle P, T, V \rangle$ besteht aus einer Menge P von *Plätzen*, einer Menge T von *Transitionen* und einer Flussfunktion $V \in \text{BAGS}((P \times T) \cup (T \times P))$, wobei $P \cap T = \emptyset$. Eine Multimenge $m \in \text{BAGS}(P)$ ist eine *Markierung* von S .

Beispiel 14

Abbildung 6 zeigt eine Netzstruktur: P_1 und P_2 sind Plätze, t_1, \dots, t_3 sind Transitionen. Ist bei einer Kante kein Gewicht angegeben, beträgt das Gewicht 1. Die Multimenge \mathcal{P}_1 ist eine Markierung von Net_6 , die Platz

Abbildung 6 Eine Netzstruktur Net_6

P_1 eine Marke und Platz P_2 keine Marke zuweist.

Eine Transition t ist *aktiviert* in einer Markierung m , falls in m ausreichend viele Marken auf den Plätzen im Vorbereich von t vorliegen. Dabei liegen auf Platz p ausreichend viele Marken, falls das Gewicht der Kante von p nach t die Anzahl der Marken auf p nicht überschreitet, also $V(p, t) \leq m(p)$ gilt.

Ist die Transition t in der Markierung m aktiviert, kann t *feuern*, indem sie entsprechend der Gewichte entlang der Kanten Marken von Plätzen *konsumiert* und auf Plätze *produziert*. Das Feuern einer Transition bildet einen *Schritt*. So resultiert das Feuern der Transition t_1 aus Markierung $\{P_1\}$ in der Folgemarkierung $\{P_2\}$.

Notation 15 (Aktiviert, Schritt)

Wir schreiben $m \xrightarrow{t}$, falls die Transition t in m aktiviert ist. Wir schreiben $m \xrightarrow{t} m'$, falls $m \xrightarrow{t}$ und das Feuern von t aus m in m' resultiert. Wir erweitern diese Schreibweise wie üblich: Wir schreiben $m_0 \xrightarrow{t_1 \dots t_n} m_n$ anstatt $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} m_n$. Eine Netzstruktur S ist *endlich*, falls $P \cup T$ endlich ist.

Beispiel 16

Die Transition t_1 ist in der in Abbildung 6 abgebildeten Markierung $\{P_1\}$ aktiviert, die übrigen Transitionen jedoch nicht. t_1 ist auch in jeder Markierung m mit $\{P_1\} \leq m$ aktiviert. Die Markierungen $\{P_1\}, \{P_2\}$ bilden mit t_2 den Schritt $\{P_1\} \xrightarrow{t_2} \{P_2\}$.

Ein *Zyklus* ist eine Folge sukzessive aktivierter Transitionen, die eine Markierung wieder in sich selbst überführt.

Beispiel 17

Zum Beispiel ist jede Folge aus $(t_1 t_2)^*$ ein Zyklus.

Definition 18 (Zyklus)

Sei $S = \langle P, T, V \rangle$ eine Netzstruktur, m eine Markierung und $w \in T^*$ mit $m \xRightarrow{w} m$. Dann ist w ein *Zyklus* von S .

2.7.1.2 Petrinetze

Ein *Petrinetz* ist eine Netzstruktur mit einer ausgezeichneten *Anfangsmarkierung* m . Auf diese Weise beschränken wir die Menge der Schritte und wir können die Begriffe des *Ablaufs*, der *Erreichbarkeit*, der *Beschränktheit* sowie der *Zyklenfreiheit* definieren.

Definition 19 (Petrinetz, Abläufe, Erreichbarkeit)

Ein *Petrinetz* $N = \langle S, m \rangle$ besteht aus einer Netzstruktur S und einer Markierung m von S . Wir nennen m die *Anfangsmarkierung* von N .

Sei $w \in T^*$ mit $m_1 \xRightarrow{w} m_n$ und $m_1 = m$.

- (1) w ist ein *Ablauf* von N ,
- (2) w *resultiert* in N in m_n ,
- (3) m_n ist *erreichbar* in N .
- (4) Sei $w = w_1 w_2 w_3$ und w_2 ein Zyklus von S . Dann *enthält* w den Zyklus w_2 .
- (5) Enthält w keinen Zyklus von S . Dann ist w *zyklenfrei*.

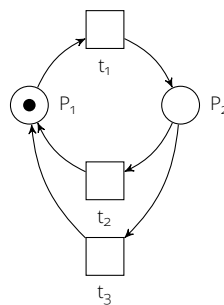


Abbildung 7 Ein Petrinetz Net_7

Beispiel 20

Abbildung 7, S. 28 zeigt ein Petrinetz. Wir können die Abläufe dieses Petrinetzes durch den regulären Ausdruck $(t_1(t_2|t_3))^*$ beschreiben. Die Menge der erreichbaren Markierungen besteht genau aus den Markierungen $\{P_1\}$ und $\{P_2\}$. Das Petrinetz ist 1-beschränkt, da in jeder erreichbaren Markierung auf jedem Platz höchstens eine Marke liegt. Das Netz ist zudem nicht zyklensfrei, da es beispielsweise den Zyklus t_1t_2 enthält.

Notation 21 (Abläufe, erreichbare Markierungen)

Sei N ein Petrinetz. Wir schreiben $\text{RUNS}(N)$ für die Menge aller Abläufe von N , $\text{RUNS}(N, m)$ für die Menge aller Abläufe von N , die in einer Markierung m resultieren, sowie $\text{RUNS}(N, M)$ für die Vereinigung der Mengen $\text{RUNS}(N, m)$ mit $m \in M$. Wir schreiben $\text{RUNS}_{\text{ZF}}(N, m)$ und $\text{RUNS}_{\text{ZF}}(N)$ für die jeweilige Restriktion der Mengen $\text{RUNS}(N, m)$ und $\text{RUNS}(N)$ auf zyklensfreie Abläufe. Wir schreiben $\text{REACH}(N)$ für die Menge aller erreichbaren Markierungen von N .

2.7.2 Rahmenmarkierungen und Zielmarkierungen

Wir betrachten häufig Netze mit zwei ausgezeichneten Markierungsmengen:

- (1) *Rahmenmarkierungen*. Eine Menge von Markierungen, die aus Sicht des Modellierers nicht verlassen werden darf.
- (2) *Zielmarkierungen*. Eine Menge von Markierungen, die aus Sicht des Modellierers besonders wünschenswert sind.

Rahmenmarkierungen

Zielmarkierungen

Weist ein Netz keine ausgezeichnete Mengen von Rahmenmarkierungen beziehungsweise Zielmarkierungen auf, betrachten wir der Einfachheit halber jede Markierung als Rahmenmarkierung sowie die Anfangsmarkierung als einzige Zielmarkierung.

2.7.3 Beschriftete Netze

In einem *beschrifteten* Netz ist jede Transition mit einem Buchstaben aus einem Alphabet oder dem leeren Wort ε beschriftet.

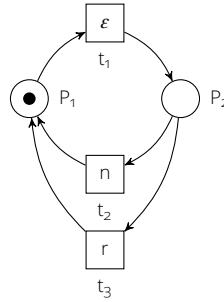


Abbildung 8 Ein beschriftetes Netz Net_8 über Alphabet $\{n, r\}$

Definition 22 (Beschriftetes Netz)

Ein *beschriftetes Netz* N über einem Alphabet Σ ist ein Netz, in dem jede Transition t mit einem Buchstaben $\ell(t) \in \Sigma \cup \{\varepsilon\}$ beschriftet ist.

Ein Ablauf $t_1 \dots t_n$ von N *erzeugt* ein Wort $w \in \Sigma^*$, falls

$$w = \ell(t_1) \dots \ell(t_n) \text{ .}$$

N *erzeugt* w , falls wenigstens ein w -erzeugender Ablauf von N existiert.

Beispiel 23

Abbildung 8 zeigt ein beschriftetes Netz. Die Inschriften der Transitionen t_1 , t_2 und t_3 sind jeweils ε , n und r . Das beschriftete Netz erzeugt somit die Wörter $(n|r)^*$.

Notation 24 (Menge der erzeugenden Abläufe für ein Wort)

Sei N ein beschriftetes Petrinetz mit Beschriftungsfunktion ℓ . Sei $v = t_1 \dots t_n \in T^*$ und $w \in \Sigma^*$. Wir notieren $\ell(t_1) \dots \ell(t_n)$ als $\ell(v)$. Wir schreiben $\text{RUNS}(N, w)$ für die Menge aller Abläufe von N , die w erzeugen.

2.7.3.1 Beziehung zu Automaten

Wir können einen Automaten A mit genau einem Anfangszustand i leicht als ein beschriftetes Netz M über dem gleichen Alphabet mit Anfangsmarkierung $\{i\}$ auffassen, indem wir für jeden Zustand einen Platz einführen und für jede Transition $e = p \xrightarrow{a} q$ von A eine Transition e einführen mit $V(p, e) = V(e, q) = 1$ und $\ell(e) = a$. Es gilt dann: $\{i\} \xrightarrow{t_1 \dots t_n}_M \{q\}$ und $w = \ell(t_1) \dots \ell(t_n)$ g.d.w. $i \xRightarrow{w}_A q$.

Wir können außerdem zu jedem beschrifteten Netz M mit Anfangsmarkierung i einen Automaten mit Anfangszustand i bilden, indem wir für jede Markierung m einen Zustand einführen und eine Transition $m_1 \xrightarrow{\ell(t)}_A m_2$ einführen, falls $m_1 \xrightarrow{t}_M m_2$.

2.7.4 Anwendungsfälle beschrifteter Netze

Ein *Anwendungsfall* [Jac+92] beschreibt ein Teilverhalten eines Systems, meist in Form von Schritten, die hintereinander eintreten müssen, damit ein fachliches Ziel erfüllt wird. In dieser Arbeit beschreiben wir einen Anwendungsfall eines beschrifteten Netzes in Form zweier Sprachen $\bullet U$ und U^\bullet . Dabei beschreibt $\bullet U$ die Bedingungen, unter denen der Anwendungsfall eintritt und U^\bullet den Hauptteil des Anwendungsfälle.

Definition 25 (Anwendungsfall)

Sei N ein Netz mit Beschriftungsfunktion $\ell: T \rightarrow \Sigma \cup \{\varepsilon\}$. Seien $L \subseteq \Sigma^*$ und $L' \subseteq \Sigma^*$ Sprachen.

- (1) L und L' bilden den *Anwendungsfall* U von N mit $\bullet U := L$ und $U^\bullet := L'$.
- (2) Die Sprache $\underline{U} := \bullet U U^\bullet$ ist das *Verhalten* von U .
- (3) Die Sprache $\underline{U} \cap \{\ell(w) \mid w \in \text{Runs}(N)\}$ ist das *U -Verhalten* von N .
- (4) N *realisiert* U , falls das U -Verhalten von N nicht-leer ist.
- (5) N *überdeckt* U , falls das U -Verhalten von N das Verhalten von U ist.
- (6) U ist *regulär*, falls $\bullet U$ und U^\bullet regulär sind.

Beispiel 26

Wir betrachten das beschriftete Netz Net_8 aus Abbildung 8, S. 30 über dem Alphabet $\Sigma = \{\text{n}, \text{r}, \text{x}\}$. Wir beschreiben eine Runde, die Net_8 vom Anfangszustand wieder in den Anfangszustand bringt, mit einem Anwendungsfall U . Dazu definieren wir $U^\bullet := \{\text{n}, \text{r}\}$. Für $\bullet U$ wählen wir der Einfachheit halber Σ^* . Dann entspricht das U -Verhalten von Net_8 genau der Menge $\text{Runs}(\text{Net}_8) \setminus \{\varepsilon\}$ der nicht-leeren Abläufe von Net_8 . Daher überdeckt Net_8 den Anwendungsfall U .

Als nächstes betrachten wir die Variante Net_8' von Net_8 , in der wir die Beschriftung von t_1 von ε auf x ändern. Auch Net_8' überdeckt den oben definierten Anwendungsfall U . Betrachten wir dagegen den Anwendungsfall U' mit $U'^\bullet = \{\text{n}\}^*$ und $\bullet U' = \Sigma^*$, beobachten wir, dass Net_8' den Anwen-

dungsfall U' zwar realisiert, jedoch *nicht* überdeckt: Das Verhalten von U' ist genau die Menge $\Sigma^*\{n\}^*$, Abläufe von Net_g' enden jedoch stets mit höchstens einem n , niemals jedoch mit n^k mit $k > 1$.

2.7.5 Offene Netze

Ein *offenes Netz* [KMR02] ist ein Petrinetz mit einer ausgezeichneten *Schnittstelle* für den Austausch von Nachrichten. Da wir ausschließlich asynchronen Austausch von Nachrichten betrachten, besteht eine Schnittstelle stets aus einer Menge von Plätzen, über die Nachrichten empfangen oder gesendet werden können. Der tatsächliche Nachrichtenaustausch wird durch die *Komposition* (Abschnitt 7.1, S. 137) realisiert.

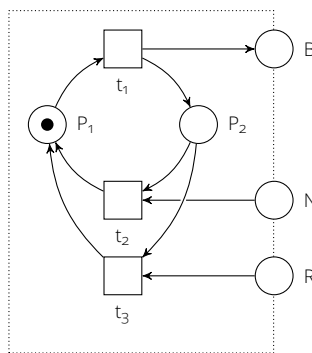


Abbildung 9 Ein offenes Netz Net_g mit Eingabepätzen N und R sowie Ausgabeplatz B

Definition 27 (Schnittstelle, offenes Netz)

Sei N ein Netz mit Plätzen P . Eine Menge $I \subseteq P$ ist eine *Schnittstelle* von N . Ein Netz mit einer ausgezeichneten Schnittstelle heißt *offenes Netz*.

Die Plätze eines offenen Netzes können wir in Ein-, Ausgabe- und interne Plätze unterteilen. Eine Transition sendet, falls sie Marken auf wenigstens einen Ausgabeplatz produziert, und empfängt, falls sie Marken von wenigstens einem Eingabeplatz empfängt:

Definition 28 (Ein-/Ausgabeplatz, Senden, Empfangen)

Sei N ein offenes Netz mit Plätzen P , Transitionen T und Schnittstelle I .

- (1) Ein Platz $p \in I$ ist ein *Eingabeplatz*, falls $V|_{\{p\} \times T} \neq \emptyset$.
- (2) Ein Platz $p \in I$ ist ein *Ausgabeplatz*, falls $V|_{T \times \{p\}} \neq \emptyset$.
- (3) Eine Transition $t \in T$ *sendet* über p , falls $p \in I$ und $V(\langle p, t \rangle) \geq 1$ gilt.
- (4) Eine Transition $t \in T$ *empfängt* über p , falls $p \in I$ und $V(\langle t, p \rangle) \geq 1$ gilt.

Beispiel 29

Abbildung 9, S. 32 zeigt ein offenes Netz mit Eingabeplätzen N und R und Ausgabeplatz O .

Ein offenes Netz ist in *Normalform* [LMW07], falls jeder Platz der Schnittstelle entweder ein Ein- oder ein Ausgabeplatz ist.

Definition 30 (Normalform für offene Netze)

Sei N ein offenes Netz mit Plätzen P , Transitionen T und Schnittstelle I , sodass

- (1) jeder Platz $p \in I$ entweder ein Eingabe- oder ein Ausgabeplatz ist, und
- (2) für jede Transition $t \in T$ gilt:

$$\sum_{p \in I} V(\langle p, t \rangle) + V(\langle t, p \rangle) \leq 1 \quad . \quad (20)$$

Dann heißt N *normal*.

Für ein offenes Netz L ist die *universelle Umgebung* $\text{ENV}(L)$ von L das (bis auf Isomorphie) kleinste Netz, das die folgende Bedingung erfüllt: Sei p ein Eingabeplatz (Ausgabeplatz) von L . Dann ist p ein Ausgabeplatz (Eingabeplatz) von $\text{ENV}(L)$ und es existiert eine Transition t die eine Marke auf p produziert (von p konsumiert) und keine weiteren Marken konsumiert oder produziert.

2.7.6 Simulation

In diesem Abschnitt wiederholen wir die Definitionen und Notationen zur aus der Literatur wohlbekannten *Simulationsrelation* [Mil89; BKO8] auf Verhalten.

Eine *Simulationsrelation* R ist eine binäre Relation auf Markierungen. Intuitiv heißt $\langle m_1, m_2 \rangle \in R$, dass in m_1 wenigstens so viel möglich ist wie in m_2 .

Definition 31 (Simulationsrelation)

Sei S eine Netzstruktur. Sei R eine binäre Relation auf den Markierungen von S , sodass für alle $\langle m_1, m_2 \rangle \in R$ und Schritte $m_1 \xrightarrow{t_1} m'_1$ eine Transition t_2 und eine Markierung m'_2 existiert mit

- (1) $m_2 \xrightarrow{t_2} m'_2$.
- (2) $\langle m'_1, m'_2 \rangle \in R$.
- (3) t_1 und t_2 haben dieselbe Beschriftung.
- (4) t_1 und t_2 senden (empfangen) jeweils die gleichen Nachrichten.
- (5) m_1 ist eine Rahmenmarkierung (Zielmarkierung) g.d.w. m_2 eine Rahmenmarkierung (Zielmarkierung) ist.

Dann ist R eine *Simulationsrelation* von S .

Unter den Simulationsrelationen auf einer Netzstruktur gibt es stets eine *maximale Simulationsrelation*, da die Vereinigung zweier Simulationsrelationen erneut eine Simulationsrelation ergibt.

In der Definition ist R eine binäre Relation über Markierungen derselben Netzstruktur. Wir können zwei Netzstrukturen S_1 und S_2 jedoch stets durch „nebeneinander schreiben“ in eine einzige Netzstruktur S_1S_2 zusammenfassen, wobei wir annehmen, dass S_1 und S_2 disjunkt sind. S_1S_2 ist also die Netzstruktur $\langle P, T, V \rangle$ mit $P = P_1 \cup P_2$, $T = T_1 \cup T_2$ und $V(x, y) = V_i(x, y)$ für alle $x, y \in (P_i \times T_i) \cup (T_i \times P_i)$ und $i = 1, 2$. Dies erlaubt es uns auch Markierungen zweier disjunkter Netzstrukturen zu vergleichen.

Insbesondere ist dies auch mit Netzen möglich. Daraus ergibt sich der Begriff der *initialen Simulationsrelation*.

Definition 32 (Initiale Simulationsrelation)

Für $i = 1, 2$, sei N_i ein Netz mit Netzstruktur S_i und Anfangsmarkierung m_i . Sei R eine Simulationsrelation auf S_1S_2 . Sei $\langle m_1, m_2 \rangle \in R$.

Dann ist R initial bezüglich N_1 und N_2 .

Ist die maximale Simulationsrelation auf S_1S_2 initial bezüglich N_1 und N_2 , dann sagen wir, dass N_1 durch N_2 *simuliert* wird.

Definition 33 (simuliert)

Für $i = 1, 2$, sei N_i ein Netz mit Netzstruktur S_i . Sei die maximale Simulationsrelation auf $S_1 S_2$ initial bezüglich N_1 und N_2 .

Dann wird N_1 durch N_2 *simuliert*, geschrieben $N_1 \leq_{\text{SIM}} N_2$.

Es ist wohlbekannt, dass die Beziehung \leq_{SIM} eine Quasiordnung auf Netzen darstellt. Weiterhin ist der folgende Zusammenhang zwischen \leq_{SIM} und den Ablaufmengen bekannt:

$$N_1 \leq_{\text{SIM}} N_2 \text{ impliziert } \text{RUNS}(N_1) \subseteq \text{RUNS}(N_2) . \quad (21)$$

Dies lässt sich per Induktion über den Aufbau der Abläufe zeigen. Die Umkehrung von Gleichung (21) gilt jedoch im Allgemeinen nicht.

Definition 34 (Bisimulation, bisimilar)

Für $i = 1, 2$, sei N_i ein Netz mit Netzstruktur S_i . Sei R eine Simulationsrelation auf $S_1 S_2$. Sei R^{-1} ebenfalls eine Simulationsrelation auf $S_1 S_2$.

Dann ist R eine Bisimulationsrelation.

Ist R initial bezüglich N_1 und N_2 , sind N_1 und N_2 bisimilar.

Es ist bekannt, dass die Bisimilarität auf Netzen eine Äquivalenzrelation ist.

2.7.7 Synchrones Produkt

Das *synchrone Produkt* N zweier Petrinetze N_1 und N_2 ist wieder ein Petrinetz, in dem jeder Schritt von N entweder ein Schritt von N_1 , ein Schritt von N_2 oder ein gemeinsamer, synchron ausgeführter Schritt von N_1 und N_2 ist. Als Parameter des Produkts geben wir eine Menge $T_{\text{sync}} \subseteq T_1 \times T_2$ an, die angibt, welche Transitionen von N_1 mit welchen Transitionen aus N_2 synchronisiert werden. Jede Transition aus N ist entweder eine Transition von einem der N_i oder eine synchronisierte Transition $\langle t_1, t_2 \rangle$. Der Effekt einer solchen Transition ist die platzweise Summe der Effekte der Transitionen.

Für $i = 1, 2$ sei $N_i = \langle P_i, T_i, V_i, m_i \rangle$ ein Netz. Seien die Komponenten von N_1 und N_2 paarweise disjunkt. Sei $T_{\text{sync}} \subseteq T_1 \times T_2$. Dann ist das *synchrone Produkt* von N_1 und N_2 bezüglich T_{sync} das folgende Netz $N = \langle P, T, V, m \rangle$:

$$(1) \ P = P_1 \cup P_2,$$

$$(2) \ T = T_1 \cup T_2 \setminus (T'_1 \cup T'_2) \cup T_{\text{sync}}, \text{ wobei } T'_1 = \{t \mid \langle t, t' \rangle \in T_{\text{sync}}\}, T'_2 = \{t \mid \langle t' \rangle \in T_{\text{sync}}\},$$

- (3) $V(p, t) = V_1(p, t) + V_2(t, p)$ und $V(t, p) = V_1(t, p) + V_2(t, p)$ für alle $p \in P$ und $t \in T \setminus T_{\text{sync}}$. $V(\langle t_1, t_2 \rangle, p) = V_1(t_1, p) + V_2(t_2, p)$ und $V(p, \langle t_1, t_2 \rangle) = V_1(p, t_1) + V_2(p, t_2)$ für alle $\langle t_1, t_2 \rangle \in T_{\text{sync}}$.
- (4) $m = m_1 + m_2$.
- (5) Die Mengen der Rahmenmarkierungen und Zielmarkierungen von N bilden wir durch Vereinigung der entsprechenden Mengen.
- (6) Eventuelle Beschriftungen und Zuweisungen von Plätzen zur Schnittstelle bleiben erhalten.

Lemma 35 (Synchrones Produkt und Bisimulation)

Sei für jede Transition $t_1 \in T_1$ in jeder von N_2 erreichbaren Markierung eine Transition t_2 aktiviert mit $\langle t_1, t_2 \rangle \in T_{\text{sync}}$ oder gelte $t'_1 \neq t_1$ für alle $\langle t'_1, t'_2 \rangle \in T_{\text{sync}}$. Sei N_2 geschlossen und existiere für jede Transition $t_2 \in T_2$ wenigstens eine Transition $t_1 \in T_1$ mit $\langle t_1, t_2 \rangle \in T_{\text{sync}}$. Dann sind N und N_1 bisimilar.

Beweis (Lemma 35)

- (1) Sei R die Menge aller $\langle m, m' \rangle$ aus Markierungen von N_1 und N mit $m = m'|_{P_1}$, m ist erreichbar in N_1 und $m'|_{P_2}$ ist erreichbar in N_2 .
- (2) Wir zeigen, dass R eine Simulation ist: Sei $\langle m_1, m'_1 \rangle \in R$ und $m_1 \xrightarrow{t_1}_{N_1} m_2$.
 - Fall 1: $t'_1 \neq t_1$ für alle $\langle t'_1, t'_2 \rangle \in T_{\text{sync}}$. Dann ist t_1 auch in m'_1 aktiviert, da $m'_1|_{P_1} = m_1$, und es gilt $m'_1 \xrightarrow{t_1}_N m'_2$ mit $m'_2|_{P_1} = m_2$. Da $m'_1|_{P_2} = m'_2|_{P_2}$ gilt, ist $m'_2|_{P_2}$ in N_2 erreichbar. Also gilt $\langle m_2, m'_2 \rangle \in R$.
 - Fall 2: in jeder von N_2 erreichbaren Markierung ist eine Transition t_2 aktiviert mit $\langle t_1, t_2 \rangle \in T_{\text{sync}}$. Da $\langle m_1, m'_1 \rangle \in R$ gilt, gilt auch $m'_1|_{P_2}$ ist in N_2 erreichbar. Also ist in $m'_1|_{P_2}$ eine Transition t_2 aktiviert mit $\langle t_1, t_2 \rangle \in T_{\text{sync}}$. Dann ist t_2 auch in m'_1 aktiviert. Also ist $\langle t_1, t_2 \rangle$ auch in m'_1 aktiviert, und es gilt $m'_1 \xrightarrow{\langle t_1, t_2 \rangle} m'_2$ für eine Markierung m'_2 . Aus $m'_1|_{P_1} = m_1$, $m_1 \xrightarrow{t_1} m_2$ und $m'_1 \xrightarrow{\langle t_1, t_2 \rangle} m'_2$ können wir $m_2 = m'_2|_{P_1}$ folgern. Außerdem ist $m'_2|_{P_2}$ in N_2 erreichbar, da $m_2|_{P_2}$ erreichbar ist und $m_2|_{P_2} \xrightarrow{t_2} m'_2|_{P_2}$ gilt. Also gilt $\langle m_2, m'_2 \rangle \in R$. Da N_2 geschlossen ist, manipuliert t_2 keine Schnittstellenplätze. Also manipuliert $\langle t_1, t_2 \rangle$ dieselben Schnittstellenplätze wie t_1 .

Also ist R eine Simulation.

(3) Wir zeigen, dass R^{-1} eine Simulation ist. Sei $\langle m_1, m'_1 \rangle \in R$ und $m'_1 \xrightarrow{t}_N m'_2$.

- $t \in T_1$. Dann ist t auch in m_1 aktiviert da $m'_1|_{P_1} = m_1$, und es gilt $m_1 \xrightarrow{t}_{N_1} m_2$ für ein m_2 . Außerdem gilt offenbar $m_1 \xrightarrow{t}_{N_1} m'_2|_{P_1}$. Außerdem gilt $m'_1|_{P_2} = m'_2|_{P_2}$ und daher ist auch $m'_2|_{P_2}$ in N_2 erreichbar. Also gilt $\langle m_2, m'_2 \rangle \in R$.
- $t \notin T_1$. Dann gilt $t = \langle t_1, t_2 \rangle$ mit $t_i \in T_i$ für $i = 1, 2$. Da t in m'_1 aktiviert ist, ist t_1 auch in m_1 aktiviert, und es gilt $m_1 \xrightarrow{t_1}_{N_1} m_2$ für eine Markierung m_2 . Es gilt $m'_2|_{P_1} = m_2$. Da m_1 erreichbar ist, ist m_2 auch erreichbar. Da t in m_1 aktiviert ist, ist auch t_2 in $m'_1|_{P_2}$ aktiviert, und es gilt $m'_1|_{P_2} \xrightarrow{t_2}_{N_2} m'_2|_{P_2}$. Da $m'_1|_{P_2}$ in N_2 erreichbar ist, ist $m'_2|_{P_2}$ ebenfalls erreichbar in N_2 . Also ist $\langle m_2, m'_2 \rangle \in R$.

Also ist R eine Bisimulation.

(4) Wir zeigen, dass R initial ist: Die initiale Markierung m_1 von N_1 ist erreichbar. Für die initiale Markierung $m = m_1 + m_2$ von N gilt: $m|_{P_2} = m_2$ ist die initiale Markierung von N_2 . Also ist m_2 erreichbar. Außerdem sieht man leicht, dass $m|_{P_1} = m_1$ gilt. Demnach gilt $\langle m_1, m \rangle \in R$.

Also ist R eine initiale Bisimulation und N_1 und N sind bisimilar.

3 Gewichtete Netze

Wir modellieren Systeme als *gewichtete Netze* [Sür12], spezielle Petrinetze in denen jede Transition eindeutige *Kosten* aufweist. Petrinetze eignen sich zur Modellierung von Struktur und funktionalen Aspekten des Verhaltens eines Systems. Mithilfe der Kosten können wir zusätzlich nicht-funktionale Aspekte beschreiben, wie beispielsweise monetäre Kosten, Energieverbrauch, Ausführungsdauer oder Zuverlässigkeit einer Transition. Durch Festlegung einer *Perspektive* und eines *Kostenmodells* definieren wir die Semantik gewichteter Netze: Eine Perspektive legt fest, welcher Teil des Verhaltens betrachtet werden soll, ein Kostenmodell beschreibt die Verrechnung einzelner Kosten zu aggregierten Kostenwerten.

Im Folgenden beschreiben wir die Syntax gewichteter Netze in Abschnitt 3.1. Anschließend führen wir *Perspektiven* in Abschnitt 3.2, S. 41 und *Kostenmodelle* in Abschnitt 3.3, S. 46 ein. Darauf aufbauend definieren wir die Semantik gewichteter Netze in Abschnitt 3.4, S. 66 und betrachten eine Klasse von Eigenschaften gewichteter Netze, die wir als *Kosteneindeutigkeit* bezeichnen in Abschnitt 3.5, S. 70. Schließlich klassifizieren wir gewichtete Netze in Abschnitt 3.6, S. 86 anhand von *Anforderungen* und in Abschnitt 3.7, S. 91 anhand *Präferenzen*. Schließlich diskutieren wir den Formalismus der gewichteten Netze in Hinblick auf verwandte Arbeiten in Abschnitt 3.8, S. 92.

3.1 Syntax gewichteter Netze

Ein *gewichtetes Netz* ist ein Petrinetz, indem jede Transition eindeutige Kosten aufweist. Diese Kosten stammen aus einer möglicherweise unendlichen Domäne \mathbb{K} .

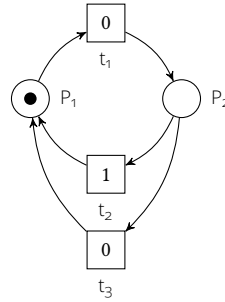
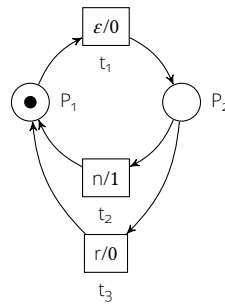
Definition 36 (Gewichtetes Netz)

Sei N ein Netz mit Transitionen T . Sei \mathbb{K} eine Menge. Sei in N jeder Transition $t \in T$ ein Wert $\llbracket t \rrbracket_N \in \mathbb{K}$ zugeordnet.

Dann ist N ein *gewichtetes Netz* und $\llbracket \cdot \rrbracket_N$ die *Kostenfunktion* von N .

Beispiel 37

Das Netz Net_{10} aus Abbildung 10 ist gewichtet über \mathbb{N}_0 : Die Transitionen

Abbildung 10 Ein gewichtetes Netz Net_{10} Abbildung 11 Ein gewichtetes, beschriftetes Netz Net_{11}

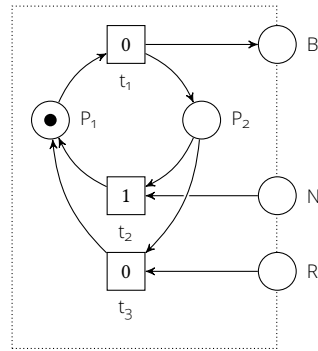
t_1 und t_3 weisen Kosten 0 auf, die Transition t_2 dagegen Kosten 1. Das Netz Net_{11} aus Abbildung 11 ist sowohl beschriftet wie gewichtet. In der graphischen Darstellung schreiben wir in eine Transition erst die Beschriftung und nach einem Schrägstrich die Kosten. Das Netz in Net_{12} aus Abbildung 12 ist offen und gewichtet.

Die Kostenfunktion $\llbracket \cdot \rrbracket_N$ transformiert eine Folge von Transitionen kanonisch in eine Folge von Werten aus D . So entsteht beispielsweise aus der Folge $t_1 \ t_2 \ t_1 \ t_3$ die Folge $0 \ 1 \ 0 \ 0$. Auf dieselbe Weise können wir $\llbracket \cdot \rrbracket_N$ auch auf Mengen und Familien übertragen.

Notation 38 (Übertragung Kostenfunktion)

Sei $t_1 \dots t_n \in T^*$, $W \subseteq T^*$ und $f = (w_i)_{i \in I} \in \text{FAM}_I(T^*)$. Wir notieren $\llbracket t_1 \rrbracket_N \dots \llbracket t_n \rrbracket_N$ als $\llbracket t_1 \dots t_n \rrbracket_N$, $\{\llbracket w \rrbracket_N \mid w \in W\}$ als $\llbracket W \rrbracket_N$, sowie $(\llbracket w_i \rrbracket_N)_{i \in I}$ als $\llbracket f \rrbracket_N$.

Häufig sind wir an den in einem gewichteten Netz vorkommenden Kostenwerten interessiert. Daher führen wir die folgende Notation ein:

Abbildung 12 Ein gewichtetes, offenes Netz Net_{12} **Notation 39 (Kostenwerte eines gewichteten Netzes)**

Sei N ein gewichtetes Netz über \mathbb{K} mit Transitionen T . Dann schreiben wir $\mathbb{K}[N]$ für die Menge $\{\llbracket t \rrbracket_N \mid t \in T\}$.

Die Semantik solch einer Folge von Werten aus D definieren wir durch ein *Kostenmodell* in Abschnitt 3.3, S. 46.

Abgrenzung zu beschrifteten Netzen. Der technische Unterschied zwischen einer Kostenfunktion und einer Beschriftungsfunktion ist subtil: Eine Beschriftungsfunktion bildet jede Transition auf einen Buchstaben aus einem endlichen Alphabet ab. Eine Kostenfunktion bildet jede Transition auf einen Wert aus einer möglicherweise unendlichen Domäne ab. Der Unterschied wird daher vor allem bei unendlichen Netzstrukturen deutlich. Semantisch verstehen wir unter einer Beschriftungsfunktion jedoch etwas fundamental anderes als unter einer Kostenfunktion: Die Beschriftungsfunktion beschreibt, welche Aktivität sich hinter der Transition verbirgt, die Kostenfunktion beschreibt nicht-funktionale Aspekte.

3.2 Perspektiven

Eine *Perspektive* legt fest, welcher Teil des Verhaltens eines gewichteten Netzes für die Bestimmung der Kosten aus Sicht des Modellierers betrachtet werden sollen. Auf diese Weise kann der Modellierer bewusst das Verhalten auf bestimmte Abläufe beschränken oder in den Abläufen nur die Kosten bestimmter Transitionen betrachten. Auf diese Weise erhält der Modellierer auch bei großen und *komponierten* Modellen (siehe Abschnitt 7.1, S. 137) die Möglichkeit, die Kosten einer Komponente oder eines bestimmten Szenarios zu betrachten. Also sind Perspektiven ein Konzept, das mit der Trennung von Belangen (engl. Separation of Concerns) verbunden ist.

Perspektive

Beispiel 40

Wir betrachten einige Beispiele für Perspektiven informal. Dazu betrachten wir das Netz aus Abbildung 11, S. 40.

- (1) *Terminierung*: Betrachte alle terminierenden Abläufe, beschränke also die Menge der Abläufe auf jene, die in einer Zielmarkierung resultieren. Sei die Anfangsmarkierung die einzige Zielmarkierung. Dann wählt die Terminierungs-Perspektive die Wörter aus, die den folgenden regulären Ausdruck erfüllen: $(t_1(t_2|t_3))^*$.
- (2) *Transitions-Filterung*: Nur bestimmte Transitionen werden betrachtet, Abläufe werden auf diese Transitionen projiziert. Wählen wir beispielsweise die Transitionen t_1 und t_2 aus, dann wählt die Filterungs-Perspektive alle Präfixe der Wörter aus, die den folgenden regulären Ausdruck erfüllen: $(t_1(t_2|\varepsilon))^*$.
- (3) *Wort-Filterung*: Nur Abläufe, die ein bestimmtes Wort bilden, werden betrachtet. Sei als Wort beispielsweise nn gewählt, dann wählt die Wort-Filterung genau die Wörter aus, die den folgenden regulären Ausdruck erfüllen: $t_1 t_2 t_1 t_2 (t_1|\varepsilon)$.
- (4) *Anwendungsfall-Betrachtung*: Betrachte nur die Ablaufstücke, die einen bestimmten Anwendungsfall abdecken. Hier wird sowohl auf der Menge der Abläufe selektiert und außerdem in einem Ablauf projiziert. Sei der Anwendungsfall U wie folgt definiert: Sei $\bullet U$ die Menge aller Wörter, die auf t_1 endet. Sei U^\bullet die Menge $(t_2|t_3)^*$. Dann wählt die zugehörige Anwendungsfall-Perspektive die Wörter aus, die den folgenden regulären Ausdruck erfüllen, aus: $t_2|t_3$.

Formal bildet eine Perspektive jede Familie nicht-leerer Transitionsfolgen auf eine neue Familie nicht-leerer Transitionsfolgen ab.

Definition 41 (Perspektive)

Sei N ein Petrinetz mit Transitionen T . Eine Funktion $\pi: \text{FAM}(T^+) \rightarrow \text{FAM}(T^+)$ ist eine *Perspektive* von N , falls π eine endliche Familie stets auf eine endliche Familie abbildet.

Wir formalisieren die oben genannten Beispiele für Perspektiven.

Notation 42 (Ausgewählte Perspektiven)

Wir formalisieren ausgewählte Perspektiven:

- (1) *Terminierung*: Sei N ein Netz mit Zielmarkierungen M . Dann schreiben wir $\text{TERM}[N]$ für die Perspektive von N mit

 $\text{TERM}[N]$

$$(w_i)_{i \in I} \mapsto (w_i)_{i \in I, m \in M, w_i \in \text{RUNS}(N, m) \setminus \{\varepsilon\}} \quad (22)$$

- (2) *Transitions-Filterung*: Sei N ein Netz mit Transitionen T und sei $T' \subseteq T$. Dann schreiben wir $\text{FILTER}[T']$ für die Perspektive von N mit

 $\text{FILTER}[T']$

$$(w_i)_{i \in I} \mapsto (w_i|_{T'})_{i \in I, w_i \notin (T \setminus T')^*} \quad (23)$$

- (3) *Wort-Filterung*: Sei N ein beschriftetes Netz über Alphabet Σ mit Beschriftungsfunktion ℓ . Sei $w \in \Sigma^*$. Dann schreiben wir

$$\text{WFILTER}[N, w]$$

 $\text{WFILTER}[N, w]$

für die Perspektive von N mit

$$(w_i)_{i \in I} \mapsto (w_i)_{i \in I, \ell(w_i) = w} \quad (24)$$

- (4) *Anwendungsfall-Betrachtung*: Sei N ein Netz und U ein Anwendungsfall von N . Dann schreiben wir $\text{AF}[N, U]$ für die Perspektive von N mit

 $\text{AF}[N, U]$

$$(w_i)_{i \in I} \mapsto (v)_{i \in I, w_i = uv, \ell(u) \in \bullet U, \ell(v) \in U^\bullet, v \neq \varepsilon} \quad (25)$$

- (5) *Zyklenfreie Abläufe*: Sei N ein Netz. Dann schreiben wir $\text{ZF}[N]$ für die Perspektive von N mit

 $\text{ZF}[N]$

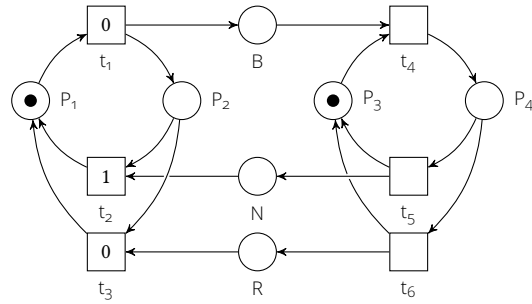
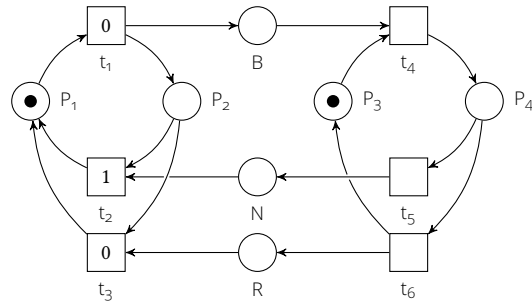
$$(w_i)_{i \in I} \mapsto (w_i)_{i \in I, w_i \in \text{RUNS}_{\text{ZF}}(N)} \quad (26)$$

Wir wenden eine Perspektive häufig auf die Menge der Abläufe eines Netzes an. Daher führen wir die folgende Notation ein.

Notation 43 (Anwendung einer Perspektive auf ein Netz)

Sei N ein Netz und π eine Perspektive von N . Sei $f = (w)_{w \in \text{RUNS}(N) \setminus \{\varepsilon\}}$. Wir notieren die Familie $\pi(f)$ auch als $\pi(N)$.

Sei N ein gewichtetes Netz mit Transitionen T und $t \in T$, sodass für alle von uns betrachteten Perspektiven π gilt: $\pi = \text{FILTER}[T \setminus \{t\}] \triangleright \pi$. Dann lassen wir der Übersichtlichkeit halber die Kosten von t in Abbildungen weg.

Abbildung 13 Ein gewichtetes Netz Net_{13} Abbildung 14 Ein gewichtetes Netz Net_{14} **Beispiel 44**

Wir betrachten die gewichteten Netze Net_{13} aus Abbildung 13 und Net_{14} aus Abbildung 14. aus den folgenden Perspektiven ($N = \text{Net}_{13}, \text{Net}_{14}$): $\pi = \text{FILTER}[t_1, t_2, t_3]$ und $\pi_N = \pi \triangleright \text{TERM}[N]$ sowie einer unten definierten Anwendungsfall-Perspektive. Da jede von uns im Beispiel betrachtete Perspektive die Transitionen t_4 , t_5 und t_6 entfernt, sind diese Transitionen nicht mit Kosten beschriftet.

Wenden wir die Perspektive π auf Net_{13} an, erhalten wir die Familie

$$\begin{aligned} \pi(\text{Net}_{13}) &= (w|_{t_1, t_2, t_3})_{w \in \text{RUNS}(\text{Net}_{13}) \setminus \varepsilon} \\ &= (w|_{t_1, t_2, t_3})_w \text{ ist nicht-leerer Präfix von } v \in (t_1 t_4 (t_5 t_2 | t_6 t_3))^* \\ &= (w)_w \text{ ist nicht-leerer Präfix von } v \in (t_1 (t_2 | t_3))^* \end{aligned}$$

Wenden wir π dagegen auf Net_{14} an, erhalten wir die Familie

$$\pi(\text{Net}_{14}) = (w)|_w \text{ ist ein nicht-leerer Präfix von } (t_1 t_4 t_6 t_3)^* t_1 t_4 t_5 t_2 t_1 \quad .$$

Wir wenden die Perspektive π_N auf $N \in \{\text{Net}_{13}, \text{Net}_{14}\}$ an, wobei wir die jeweilige Anfangsmarkierung als einzige Zielmarkierung wählen. Dann

erhalten wir

$$\pi_{\text{Net}_{13}}(\text{Net}_{13}) = \pi(\text{Net}_{13}) \quad (27)$$

$$\pi_{\text{Net}_{14}}(\text{Net}_{14}) = (w)_{w \neq \varepsilon} \text{ matcht } (t_1 t_4 t_6 t_3)^* \quad . \quad (28)$$

Sei $\Sigma = \{b, n, r\}$ und U der Anwendungsfall mit $\bullet U = \Sigma^* \{n\} \Sigma^*$ und $U^\bullet = \{n\}$. Sei für $N \in \{\text{Net}_{13}, \text{Net}_{14}\}$ jeweils N' die Variante, in der die Transitionen die folgende Beschriftung aufweisen: $t_1 \mapsto b$, $t_2 \mapsto n$ und $t_3 \mapsto r$ und $t \mapsto \varepsilon$ für die übrigen Transitionen. Sei $\pi'_N = \pi \triangleright \text{AF}[N', U]$. Dann erhalten wir

$$\pi'_{\text{Net}_{13}}(\text{Net}_{13}) = (t_2)_{w \in \Sigma^* \{n\} \Sigma^* \{n\}}$$

$$\pi'_{\text{Net}_{14}}(\text{Net}_{14}) = \text{die leere Familie} \quad .$$

Eine Perspektive heißt *kompositional*, falls es keinen Unterschied macht, ob sie schrittweise auf Teilfamilien über disjunkten Indexmengen oder direkt auf die gesamte Familie angewendet wird.

Definition 45 (kompositional)

Sei N ein Netz mit Transitionen T . Sei π eine Perspektive über N , sodass für Familien $f \in \text{FAM}(T^+)$ und Zerlegungen $f = f_1 \cup f_2$ von f über disjunkten Indexmengen gilt:

- (1) $\text{DEF}(\pi(f_1))$ und $\text{DEF}(\pi(f_2))$ sind disjunkt und
- (2) $\pi(f) = \pi(f_1) \cup \pi(f_2)$.

Dann ist π *kompositional*.

Lemma 46

Jede in Notation 42, S. 43 beschriebene Perspektive ist kompositional.

Beweis (Lemma 46)

Jede der Perspektiven kann durch eine Menge $P \subseteq T^+$ sowie eine Funktion $f: T^+ \rightarrow T^+$ beschrieben werden, sodass gilt $(w_i)_{i \in I} \mapsto (f(w_i))_{i \in I, w_i \in P}$. Seien f, f_1 und f_2 wie in Definition 45 definiert.

- (1) Für $i = 1, 2$ gilt: $\text{DEF}(\pi(f_i)) \subseteq \text{DEF}(f_i)$. Da $\text{DEF}(f_1)$ und $\text{DEF}(f_2)$ disjunkt sind, sind es auch die Teilmengen $\text{DEF}(\pi(f_1))$ und $\text{DEF}(\pi(f_2))$.
- (2) Da π nur entsprechend P auswählt und mit f transformiert, gilt auch $\pi(f) = \pi(f_1) \cup \pi(f_2)$.

Analog zu Notation 39, S. 41 führen wir die folgende Notation ein:

Notation 47 (Kostenwerte eines gewichteten Netzes)

Sei N ein gewichtetes Netz über \mathbb{K} mit Transitionen T und Kostenfunktion $\llbracket \cdot \rrbracket_N$. Sei π eine Perspektive von L . Dann schreiben wir $\mathbb{K}[N, \pi]$ für die Menge der $\llbracket t \rrbracket_N$ mit $t \in T$, $f \in \text{FAM}(T^+)$, $\pi(f) = (a_i)_{i \in I}$, $a_i = t_1 \dots t_n$, $t_j = t$, $i \in I$, $1 \leq j \leq n$.

3.3 Kostenmodelle

In diesem Abschnitt definieren wir *Kostenmodelle* zur Verrechnung von Kosten. Als eine wichtige Unterklasse betrachten wir *Kostenmodelle über vollständigen Verbänden* (siehe Abschnitt 3.3.1, S. 49) und führen darauf basierend den Begriff der *Schwelle* eines Kostenmodells ein (siehe Abschnitt 3.3.2, S. 50). Wir identifizieren wichtige Eigenschaften von Kostenmodellen (siehe Abschnitt 3.3.3, S. 51), die wir im Verlauf dieser Arbeit immer wieder aufgreifen. Wir führen den Begriff des *Abschlusses* eines Kostenmodells ein (siehe Abschnitt 3.3.4, S. 53) und diskutieren die mögliche Visualisierung durch Graphen eines Kostenmodells (siehe Abschnitt 3.3.5, S. 54). Schließlich führen wir das *Kreuzprodukt* als Kompositionsoperator auf Kostenmodellen ein (siehe Abschnitt 3.3.6, S. 60).

Kostenmodell

Ein *Kostenmodell* legt Funktionen zur Verrechnung von Kosten fest. Dabei orientieren wir uns grob an der Definition 11, S. 24 *partiell vollständiger Bewertungs-Monoide* [DM11] aus dem Forschungsgebiet der gewichteten Automaten.

Beispiel 48

Beispiele für Funktionen zur Verrechnung von Kosten sind:

- Einfache, binäre Rechenoperatoren auf Zahlen wie Summe, Multiplikation
- Komplexere Operatoren auf Zahlen wie das arithmetische oder geometrische Mittel oder Discounted Sum,
- Vergleichsoperatoren, wie Infimum oder Supremum.

Betrachten wir ein gewichtetes Netz aus einer Perspektive, erhalten wir eine Familie von Transitionsfolgen. Durch die Anwendung der Kostenfunktion des Netzes erhalten wir eine Familie von Folgen von Kostenwerten. Die Aufgabe eines Kostenmodells besteht darin, diese Familie auf einen einzelnen Wert abzubilden. In unserer Arbeit teilen wir – analog zu gewichteten Automaten – diese Aufgabe in zwei Schritte auf:

- (1) Verrechne eine endliche, nicht-leere Folge von Kosten.

- (2) Verrechne eine Familie von Kosten.

Die Aufgabe, eine Familie von Folgen von Kosten auf einen Wert abzubilden, geschieht dann in zwei Schritten: Erst wird jede Folge in einen Wert umgewandelt, was in einer Familie von Werten resultiert. Dann wird die Familie von Werten zu einem einzelnen Wert verrechnet.

Ein Kostenmodell legt daher zur Lösung dieser Aufgaben eine Funktion zur Verrechnung einer Folge sowie eine Funktion zur Verrechnung einer Familie fest.

Definition 49 (Kostenmodell)

Sei \mathbb{K} eine Menge. Sei $\prod: \mathbb{K}^+ \rightarrow \mathbb{K}$ mit $\prod(a) = a$ für alle $a \in D$. Sei \sum ein totaler Kompressor über \mathbb{K} . Sei $w \in \mathbb{K}^+$ ein Wort. Seien $f \in \text{FAM}(\mathbb{K})$ und $g = (w_i)_{i \in I} \in \text{FAM}(\mathbb{K}^+)$ Familien.

- (1) $\mu := \langle \mathbb{K}, \sum, \prod \rangle$ ist ein *Kostenmodell*.
- (2) $\mu(w) := \prod(w)$ sind die *Kosten* von w bezüglich μ .
- (3) $\mu(f) := \sum(f)$ sind die *Kosten* von f bezüglich μ .
- (4) $\mu(g) := \sum((\prod(w_i))_{i \in I})$ sind die *Kosten* von g bezüglich μ .

Beziehung zu partiell vollständigen Bewertungs-Monoiden. Einerseits sind Kostenmodelle spezieller als partiell vollständige Bewertungs-Monoiden und Multi-Hemiringe, da sie für alle Familien definiert sind. Andererseits erfüllen die Operatoren \prod und \sum im Allgemeinen nicht alle Anforderungen der Operatoren im partiell vollständigen Bewertungs-Monoiden bzw. Multi-Hemiring. Daher sind Kostenmodelle und partiell vollständige Bewertungs-Monoiden zunächst unvergleichbar; vollständige Bewertungs-Monoiden stellen eine echte Teilmenge dar.

Wir können jedoch zu jedem Bewertungs-Monoid kanonisch ein Kostenmodell bilden, indem wir es totalisieren. Dazu führen wir das Symbol ω für nicht-definierte Kosten ein. Die konkrete Wahl von ω ist im Rahmen dieser Arbeit unerheblich. Sei im Folgenden daher ω fest gewählt; für alle folgenden Bewertungs-Monoiden M nehmen wir außerdem an, dass ω nicht in der Trägermenge \mathbb{K} enthalten ist.

Notation 50 (Induziertes Kostenmodell)

Wir führen folgende Notationen für Kostenmodelle ein:

- (1) Sei $M = \langle \mathbb{K}, \Sigma, \text{val}, 0 \rangle$ ein partiell vollständiges Bewertungs-Monoid. Sei Σ' die ω -Totalisierung von Σ auf \mathbb{K} . Dann notieren wir das Kostenmodell $\langle \mathbb{K}, \Sigma', \circ^+ \cup \{ \langle a, \omega \rangle \mapsto \omega, \langle \omega, a \rangle \mapsto \omega \mid a \in \mathbb{K} \} \rangle$ auch als $\text{KM}(M)$.
- (2) Sei H ein Bewertungs-Monoid und H' die kanonische Generalisierung von H . Dann notieren wir das Kostenmodell $\text{KM}(H')$ auch als $\text{KM}(H)$.

Beispiel 51

Wir betrachten einige in dieser Arbeit wiederkehrende Kostenmodelle, die auf Multi-Hemiringen (siehe Definition 7, S. 22) basieren.

- (1) Die *Max-Plus-Algebra* $\text{MP} = \langle \mathbb{N}_0 \cup \{-\infty\}, \max, + \rangle$ induziert das Kostenmodell μ über $\mathbb{K} = \mathbb{N}_0 \cup \{-\infty, \omega\}$, wobei für ein Wort $w = a_1 \dots a_n \in \mathbb{K}^+$ gilt:
 - $\mu(w) = \omega$, falls für wenigstens ein $1 \leq i \leq n$ gilt: $a_i = \omega$. Die Kosten des Wortes $1 \ \omega \ 17 \ -\infty$ betragen beispielsweise ω .
 - $\mu(w) = -\infty$, falls für alle $1 \leq i \leq n$ gilt: $a_i \neq \omega$ und wenigstens ein $1 \leq j \leq n$ existiert mit $a_j = -\infty$. Die Kosten des Wortes $1 \ 17 \ -\infty \ 11$ betragen beispielsweise $-\infty$.
 - Ansonsten entspricht $\mu(w)$ der klassischen Summe der a_i . Die Kosten des Wortes $1 \ 17 \ 11$ betragen beispielsweise 29.

Ähnlich verhält es sich mit Familien von Elementen:

- Für die leere Familie entsprechen die Kosten $-\infty$.
 - Für alle beschränkten Familien über $\mathbb{N}_0 \cup \{-\infty\}$ entsprechen die Kosten genau dem größten Element.
 - Für alle unbeschränkten Familien und alle Familien, die ω enthalten, betragen die Kosten ω .
- (2) Der *Min-Probability-Semiring* $\langle \{r \in \mathbb{R} \mid 0 < r \leq 1\}, \min, \cdot \rangle$, mit dem beispielsweise Wahrscheinlichkeiten > 0 ausgedrückt werden können, die entlang von ω -freien Wörtern multipliziert werden.
 - (3) Der *Max-Average-Multi-Hemiring* $\langle \mathbb{R} \cup \{-\infty\}, \max, \circ \rangle$ mit $a \circ_{m,n} b = \frac{a \cdot m + b \cdot n}{m+n}$, der einem ω -freien Wort sein arithmetisches Mittel zuweist: Die Kosten eines ω -freien Wortes $w = a_1 \dots a_n$ betragen also $\frac{a_1 + \dots + a_n}{n}$.

- (4) Der *Max-Discounted-Sum-Multi-Hemiring* $\langle \mathbb{R} \cup \{-\infty\}, \max, \circ \rangle$ mit $a \circ_{m,n} b = a + \lambda^m b$, wobei $0 < \lambda < 1$ fest gewählt ist, der einem ω -freien Wort seine mit Discount-Faktor λ diskontierte Summe zuweist: Die Kosten eines ω -freien Wortes $w = a_1 \dots a_n$ betragen also $a_1 + \lambda a_2 + \lambda^2 a_3 + \dots + \lambda^{n-1} a_n$.
- (5) Der *boolesche Semiring* $\langle \{0, 1\}, \vee, \wedge \rangle$, in dem jedem Wort die Kosten 0 genau dann zugeordnet werden, wenn wenigstens eine 0 im Wort vorkommt. Alle Familien über dem booleschen Semiring sind beschränkt, da der Träger endlich ist. Daher liefert die kanonische Generalisierung bereits eine totale Summe auf beliebigen Familien: Einer Familie werden genau dann Kosten 1 zugeordnet, wenn die 1 in der Familie vorkommt.

3.3.1 Verbands-Kostenmodelle

In dieser Arbeit möchten wir geordnete Kosten betrachten, um verschiedene Kostenwerte miteinander in Relation setzen zu können. Zu diesem Zweck können wir stets eine Ordnung auf den Werten annehmen oder definieren. Es gibt jedoch auch Kostenmodelle, die eine solche Ordnung bereits selbst definieren: In diesem Abschnitt definieren wir die Klasse der *Kostenmodelle über vollständigen Verbänden* oder kurz: der *Verbands-Kostenmodelle*. Intuitiv aggregiert ein solches Kostenmodell eine Familie entsprechend einer Ordnung durch Bildung des Supremums, also der kleinsten oberen Schranke der Kosten in der Familie. Nimmt man an, dass niedrigere Kosten bevorzugt werden, wird also der *Worst Case* betrachtet. Da ein Kostenmodell jede beliebige – und damit auch leere oder unbeschränkte – Familien von Kostenwerten aggregieren kann, existieren bezüglich der Ordnung Suprema für jede beliebige Menge. Nehmen wir zum Vergleich von Kosten Halbordnungen an, ist die Ordnung ein vollständiger Verband (siehe Abschnitt 2.4.3, S. 19). In Abschnitt 3.8.5, S. 96 gehen wir auf den Unterschied zwischen der Betrachtung des Worst Cases und den Best Cases ein.

Definition 52 (Verbands-Kostenmodell)

Sei μ ein Kostenmodell über \mathbb{K} . Sei $\langle \mathbb{K}, \leq \rangle$ ein vollständiger Verband, sodass für alle Familien $X \in \text{FAM}(\mathbb{K})$ gilt: $\mu(X)$ ist das Supremum von X bezüglich \leq .

Dann ist μ ein *Kostenmodell über dem vollständigen Verband* $\langle \mathbb{K}, \leq \rangle$ oder kurz: ein *Verbands-Kostenmodell*.

Beispiel 53

Die Max-Plus-Algebra induziert ein Kostenmodell μ über dem vollständi-

gen Verband $\langle \mathbb{N}_0 \cup \{-\infty, \omega\}, \leq \rangle$ mit

$$-\infty \leq y \text{ für alle } x \in \mathbb{N}_0 \cup \{-\infty, \omega\} \quad (29)$$

$$x \leq \omega \text{ für alle } x \in \mathbb{N}_0 \cup \{-\infty, \omega\} \quad (30)$$

$$x \leq y \text{ für alle } x, y \in \mathbb{N}_0 \text{ mit } x \text{ kleiner gleich } y. \quad (31)$$

Jedes Kostenmodell ist über höchstens einem Verband definiert:

Lemma 54

Sei μ ein Kostenmodell über den vollständigen Verbänden $\langle \mathbb{K}, \leq \rangle$ und $\langle \mathbb{K}, \leq' \rangle$. Dann gilt $a \leq b$ g.d.w. $a \leq' b$ für alle $a, b \in \mathbb{K}$.

Beweis (Lemma 54)

Seien $a, b \in \mathbb{K}$. Sei f die Familie über der Indexmenge $\{1, 2\}$ mit $f(1) = a$ und $f(2) = b$. Dann gilt $\sum(f)$ ist das Supremum von a und b bezüglich \leq und bezüglich \leq' . Sei $a \leq b$. Dann ist b das Supremum von $\{a, b\}$ und es gilt $\sum(f) = b$. Also gilt auch $a \leq' b$. Die andere Richtung können wir analog zeigen.

Daher können wir den Begriff *der Ordnung eines Verbands-Kostenmodells* verwenden und die folgende Notation einführen:

Notation 55 (Ordnung eines Kostenmodells)

Sei μ ein Verbands-Kostenmodell. Dann schreiben wir auch \leq_μ für die Ordnung von μ .

3.3.2 Schwellen eines Kostenmodells

Eine *Schwelle* ist ein Wert, der — einmal überschritten — entweder nicht mehr oder immer auf die gleiche Weise unterschritten wird.

Definition 56 (Schwelle)

Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ und $X \subseteq \mathbb{K}$. Gelte für alle $x \in X$ genau eine der folgenden Eigenschaften:

- (1) Für alle $w \in X^+$ mit $\mu(w) \not\leq_\mu k$: $\mu(wx) \not\leq_\mu k$.
- (2) Es existiert ein $y \in X$, sodass für alle $w \in X^+$ mit $\mu(w) \not\leq_\mu k$: $\mu(wx) = y$.

Dann ist k eine *Schwelle* von X in μ . Ist $X = \mathbb{K}$, ist k eine Schwelle von μ .

Wir führen die folgende Notation ein:

Notation 57 (Richtung eines Wertes an einer Schwelle)

Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ und $X \subseteq \mathbb{K}$. Sei k eine Schwelle von X ein μ . Sei $x \in X$. Dann schreiben wir $k \uparrow x$, falls Fall 1 aus Definition 56, S. 50 für x gilt, und $k \downarrow_y x$, falls Fall 2 mit y gilt.

Beispiel 58

Seien μ und \leq definiert wie in Beispiel 53, S. 49.

- (1) Jedes $k \in \mathbb{N}_0$ ist eine Schwelle von $\mathbb{N}_0 \cup \{-\infty\}$ und $\mathbb{N}_0 \cup \{\omega\}$ in μ , da für alle $x \in \mathbb{N}_0 \cup \{\omega\}$ der erste Fall, und für $x = -\infty$ der zweite Fall eintritt.
- (2) Kein $k \in \mathbb{N}_0$ ist eine Schwelle von $\mathbb{N}_0 \cup \{-\infty, \omega\}$ in μ , da dann für $x = -\infty$ weder der erste, noch der zweite Fall eintreten kann: Falls $\mu(w) = \omega > k$ gilt, gilt $\mu(w - \infty) = \omega$; falls $\mu(w) > k$ jedoch $\mu(w) \neq \omega$ gilt, gilt $\mu(w - \infty) = -\infty$.
- (3) ω ist keine Schwelle beliebiger Teilmengen X in μ , da für alle w gilt: $\mu(w) \leq \omega$. Also ist für alle x sowohl der erste wie der zweite Fall erfüllt.

3.3.3 Eigenschaften von Kostenmodellen

Wir betrachten die folgenden Eigenschaften von Kostenmodellen: *Kompositionalität* in Abschnitt 3.3.3.1, *Assoziativität* in Abschnitt 3.3.3.2, *Kommutativität* in Abschnitt 3.3.3.3 und *Monotonie* in Abschnitt 3.3.3.4, S. 53. Dabei betrifft die Kompositionalität als einzige nur die Aggregation über Familien, die restlichen Eigenschaften betrachten nur die Aggregation über Folgen.

3.3.3.1 Kompositionale Kostenmodelle

Ein Kostenmodell heißt *kompositional*, falls der Kompressor zur Aggregation entlang von Familien kompositional ist. Also kann ein kompositionales Kostenmodell schrittweise auf Teilfamilien von Kostenwerten angewendet werden.

Definition 59 (Kompositional)

Sei $\mu = \langle \mathbb{K}, \sum, \prod \rangle$ ein Kostenmodell, sodass \sum kompositional ist. Dann ist μ *kompositional*.

Per Definition ist die Summe in einem partiell vollständigen Monoid ein kompositionaler Kompressor. Also ist jedes durch einen partiell vollständigen Multi-Hemiring induzierte Kostenmodell kompositional.

Lemma 60

Sei M ein partiell vollständiger Multi-Hemiring. Dann ist $KM(M)$ kompositional.

Beweis (Lemma 60)

Sei $f = (a_i)_{i \in I}$. Der Kompressor \sum ist kompositional. Die ω -Totalisierung definiert $\sum f = \omega$, falls \sum für f nicht definiert ist, oder für wenigstens ein $i \in I$ gilt: $a_i = \omega$. Im Falle, dass \sum für f nicht definiert ist und kein i mit $a_i = \omega$ existiert, ist \sum auch für Zerlegungen von f nicht definiert. Also gilt für jede Zerlegung in Teilfamilien g_j : \sum ist nicht für g_j definiert. Also gilt für jede Teilfamilie g_j , dass die ω -Totalisierung von \sum für g_j als ω definiert ist. Im Falle, dass f wenigstens ein ω enthält, also ein $i \in I$ existiert mit $a_i = \omega$, gilt für jede Zerlegung in Teilfamilien g_j mit $j \in J$, dass wenigstens ein g_j existiert, sodass g_j ein ω enthält. Also liefert die ω -Totalisierung für dieses g_j auch ω . Also liefert die ω -Totalisierung für wenigstens eine Teilfamilie ω .

3.3.3.2 Assoziative Kostenmodelle

Ein Kostenmodell heißt *assoziativ*, falls die Aggregation einer Folge stets durch schrittweise Aggregation von Teilfolgen möglich ist. Sei μ ein Kostenmodell über \mathbb{K} . Eine Menge $X \subseteq \mathbb{K}$ heißt *assoziativ*, falls für alle $w = uv \in X^+$ gilt: $\mu(w) = \mu(\mu(u)\mu(v))$.

Beispiel 61

- (1) Sei μ durch einen Hemiring induziert. Dann ist μ assoziativ, da das Produkt assoziativ ist und die Erweiterung um ω diese Assoziativität nicht verletzt.
- (2) Sei μ über den rationalen Zahlen so definiert, dass gilt:

$$\mu(a_1 \dots a_n) = \frac{a_1 + \dots + a_n}{n}.$$

Dann ist μ *nicht* assoziativ, da für die Sequenz 1 2 3 gilt: $\mu(1 \ 2 \ 3) = 2$, jedoch $\mu(\mu(1 \ 2)\mu(3)) = \mu(\frac{3}{2} \ 3) = \frac{9}{2} \neq 2$.

3.3.3.3 Kommutative Kostenmodelle

Ein *kommutatives* Kostenmodell beachtet bei der Aggregation von Werten entlang einer Folge nicht die Reihenfolge der auftretenden Werte: Sei μ ein

Kostenmodell über \mathbb{K} . Eine Menge $X \subseteq \mathbb{K}$ heißt *kommutativ*, falls für alle Permutationen $v, w \in X^+$ stets $\mu(v) = \mu(w)$ gilt. Ist \mathbb{K} in μ kommutativ, heißt μ *kommutativ*. Ist jedes Produkt eines Multi-Hemirings kommutativ, ist auch das induzierte Kostenmodell kommutativ.

3.3.3.4 Monotone Kostenmodelle

In einem *monotonen* Kostenmodell werden Verlängerungen einer Folge höchstens teurer. Daher kann Monotonie nur in Anwesenheit einer Ordnung definiert werden. Der Einfachheit halber definieren wir die Eigenschaft nur für Verbands-Kostenmodelle: Sei μ ein Verbands-Kostenmodell \mathbb{K} . Eine Menge $X \subseteq \mathbb{K}$ ist *monoton* in μ , falls für alle $v, w \in X^+$ gilt:

$$\mu(v) \leq_{\mu} \mu(vw) . \quad (32)$$

Ist \mathbb{K} in μ monoton, heißt μ monoton.

Beispiel 62

Seien μ und \leq so definiert wie in Beispiel 53, S. 49.

- (1) Die Menge $\mathbb{N}_0 \cup \{\omega\}$ ist monoton in μ , da die Addition auf \mathbb{N}_0 monoton ist und außerdem $x + \omega \geq x$ für alle $x \in \mathbb{N}_0 \cup \{\omega\}$ gilt.
- (2) Die Menge $\mathbb{N}_0 \cup \{-\infty\}$ ist dagegen nicht monoton in μ , da $x + -\infty = -\infty \leq x$ für alle $x \in \mathbb{N}_0 \cup \{-\infty\}$ gilt.

Wir beobachten den folgenden Zusammenhang zwischen Assoziativität und Monotonie: Sei X monoton und assoziativ. Dann gelten:

$$\mu(v) \leq \mu(vw) \text{ impliziert } \mu(vw) \leq \mu(vww) , \quad (33)$$

und

$$\mu(u) = \mu(uv) \text{ impliziert } \mu(uw) = \mu(uvw) . \quad (34)$$

3.3.4 Abschluss eines Kostenmodells

Der *Abschluss* einer Menge X von Werten in einem Kostenmodell μ ist die Menge der Werte, die durch Anwendung von μ auf Folgen über X entstehen kann.

Definition 63 (Abschluss)

Sei μ ein Kostenmodell über \mathbb{K} . Sei $X \subseteq \mathbb{K}$. Die Menge $(X)^{\mu} := \{\mu(w) \mid w \in X^+\}$ ist der *Abschluss* von X in μ .

$(X)^{\mu}$

Betrachten wir ein Verbands-Kostenmodell, können wir zusätzlich den k -Abschluss definieren: Der k -Abschluss einer Menge X ist die Restriktion des Abschlusses von X auf Werte kleiner gleich k .

Definition 64 (k -Abschluss)

Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ und $X \subseteq \mathbb{K}$. Die Menge $(X)_{\leq k}^\mu := \{d \mid d \in (X)^\mu, d \leq_\mu k\}$ ist der k -Abschluss von X in μ .

$(X)_{\leq k}^\mu$

3.3.5 Graphen eines Kostenmodells

Ein Graph G *visualisiert* den Abschluss einer Menge X in Kostenmodell μ , falls die Kosten einer jeden Folge $w \in X^+$ in G ablesbar sind. Formal definieren wir diesen Zusammenhang wie folgt:

Definition 65 (Graph eines Kostenmodells)

Sei μ ein Kostenmodell über \mathbb{K} , $X \subseteq \mathbb{K}$ und $c \notin \mathbb{K}$. Sei G ein Kanten- und Knoten-beschrifteter, gerichteter Graph mit Knoten V und Kanten E . Existiere genau ein Knoten v_0 ohne eingehende Kanten und seien alle Knoten von v_0 aus erreichbar. Sei $\ell: V \cup E \rightarrow \mathbb{K} \cup \{c\}$ die Beschriftungsfunktion. Habe jeder Knoten $v \in V$ für jedes $x \in X$ genau eine mit x beschriftete, ausgehende Kante. Gelte $\ell(v_0) = c$ und für alle von v_0 ausgehenden und in einem Knoten v resultierende Pfade über sukzessive Kanten $e_1 \dots e_n$:

$$e_1 \dots e_n \in X^+ \quad (35)$$

$$\mu(\ell(e_1) \dots \ell(e_n)) = \ell(v) \quad (36)$$

Dann ist G ein Graph von X in μ .

Offenbar gilt für alle Knoten $v \neq v_0$ in G :

$$\ell(v) \in (X)^\mu, \quad (37)$$

das heißt, v ist mit einem Wert aus dem Abschluss von X in μ beschriftet. Zu jedem nicht-leeren $X \subseteq \mathbb{K}$ existiert ein kanonischer, unendlicher Graph von X in μ : Wir können die Menge X^+ leicht als Baum visualisieren, die Beschriftung jedes Knotens v wird durch den einzigen Pfad von v_0 zu v determiniert. Die Existenz eines *endlichen* Graphen von X ist dagegen nicht trivial und im Allgemeinen unentscheidbar:

Lemma 66 (Existenz endlicher X -Graphen ist unentscheidbar)

Sei μ ein Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$ eine endliche Menge. Die Existenz eines endlichen Graphen von X in μ ist unentscheidbar.

Beweis (Lemma 66, S. 54)

Sei Σ ein Alphabet und $\mathbb{K} = X = \Sigma^+$. Sei L eine kontextsensitive Sprache. Sei $a \in \Sigma$ und $X \subseteq \Sigma$. Für alle $w \in X^+$ sei $\mu(w) = a$, falls $w \in L$ und $\mu(w) = aa$, sonst. Sei G ein Graph von X in μ mit Wurzelknoten v_0 . Dann können wir G leicht als Automaten A mit Initialzustand v_0 auffassen, wobei ein Knoten v genau dann ein Endzustand ist, falls $\ell(v) = a$. Offenbar akzeptiert A dann genau L . Ist A und damit auch G endlich, ist L regulär. Anders herum können wir jeden L -akzeptierenden Automaten leicht als Graphen von X in μ auffassen. Existiert solch ein endlicher Automat, ist L regulär. Also hat μ genau dann einen Graphen von X , wenn L regulär ist. Da die Regularität einer kontextsensitiven Sprache unentscheidbar ist, ist auch die Existenz eines endlichen Graphen von X in μ unentscheidbar.

Für assoziative Kostenmodelle können wir immerhin die Existenz eines Graphen von X in μ zeigen für den zwei unterschiedliche Knoten stets unterschiedlich beschriftet sind. Dazu definieren wir zunächst den *kanonischen Graphen* von X in μ .

Definition 67 (Graph assoziativer Kostenmodelle)

Sei μ ein assoziatives Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Sei $c \notin \mathbb{K}$. Wir definieren induktiv den Graphen $G_{\mu, X, c}$ mit Wurzelknoten c wie folgt:

- (1) Sei $x \in X$. Dann ist $y = \mu(x)$ ein Knoten mit Beschriftung y und es gibt genau eine mit x beschriftete Kante von c nach y .
- (2) Sei $x \neq c$ ein Knoten und $x' \in X$. Dann ist $y = \mu(x\mu(x'))$ ein Knoten von G mit Beschriftung y und es gibt genau eine mit x' beschriftete Kante von x nach y .

Lemma 68

Sei μ ein assoziatives Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Sei $c \notin \mathbb{K}$.

Der Graph $G_{\mu, X, c}$ ist ein Graph von X in μ .

Beweis (Lemma 68)

Sei w ein Pfad von $G_{\mu, X, c}$, der im Wurzelknoten c startet, in x resultiert und die Kanten $e_1 \dots e_n$ mit jeweiligen Beschriftungen $x_1 \dots x_n$ traversiert. Wir zeigen: $\mu(x_1 \dots x_n) = x$.

- (1) Sei $n = 1$. Dann folgt aus der ersten Regel:

$$\mu(x_1 \dots x_n) = \mu(x_1) = x \quad .$$

- (2) Gelte die Eigenschaft für w . Existiere eine mit y beschriftete Kante von x zu einem Knoten z . Dann gilt

$$\begin{aligned}\mu(x_1 \dots x_n y) &= \mu(\mu(x_1 \dots x_n) \mu(y)) \\ &= \mu(x \mu(y)) = z \quad .\end{aligned}$$

Also ist $G_{\mu, X, c}$ ein Graph von X in μ .

Da jeder Knoten $v \neq c$ mit v beschriftet ist, gilt für zwei Knoten $v, v' \neq c$:

$$v \neq v' \text{ impliziert } \ell(v) \neq \ell(v') \quad . \quad (38)$$

Die konkrete Wahl von c ist unerheblich. Daher können wir alle $G_{\mu, X, c}$ für ein festes μ und ein festes X miteinander identifizieren:

Notation 69 (Graph eines assoziativen Kostenmodells)

Sei μ ein assoziatives Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Wir identifizieren alle Graphen $G_{\mu, X, c}$ mit $c \notin \mathbb{K}$ und schreiben $G_{\mu, X}$ anstatt $G_{\mu, X, c}$.

Seien X und der Abschluss von X in einem assoziativen Kostenmodell μ endlich. Dann folgt aus Gleichung (38) direkt die Endlichkeit des kanonischen Graphen von X in μ :

Korollar 70

Sei μ ein assoziatives Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Seien X und $(X)^\mu$ endlich.

Dann ist $G_{\mu, X}$ endlich.

Ist μ ein Verbands-Kostenmodell, können wir den Begriff des k -Graphen von X für ein $k \in \mathbb{K}$ von μ definieren. Intuitiv können wir uns den k -Graphen als eine Visualisierung des k -Abschlusses vorstellen.

Definition 71 (k -Graph eines Kostenmodells)

Sei μ ein Verbands-Kostenmodell über \mathbb{K} und sei $k \in \mathbb{K}$. Sei $X \subseteq \mathbb{K}$ und $c, \frac{1}{2} \notin \mathbb{K}$. Sei G ein Kanten- und Knoten-beschrifteter, gerichteter Graph mit Knoten V und Kanten E . Existiere genau ein Knoten v_0 ohne eingehende Kanten und seien alle Knoten von v_0 aus erreichbar. Sei $\ell: V \cup E \rightarrow \mathbb{K} \cup \{c, \frac{1}{2}\}$ die Beschriftungsfunktion. Habe jeder Knoten $v \in V$ für jedes $x \in X$ genau eine mit x beschriftete, ausgehende Kante. Gelte $\ell(v_0) = c$ und für alle von v_0 ausgehenden und in einem Knoten v resultierende Pfade über sukzessive Kanten $e_1 \dots e_n$:

$$e_1 \dots e_n \in X^+ \quad (39)$$

$$\ell(v) = \begin{cases} \mu(\ell(e_1) \dots \ell(e_n)), & \text{falls } \mu(\ell(e_1) \dots \ell(e_n)) \leq_\mu k, \\ \frac{1}{2}, & \text{sonst.} \end{cases} \quad (40)$$

Dann ist G ein k -Graph von X in μ .

Da ein kanonischer, unendlicher Graph von X in μ für alle nicht-leeren X existiert, existiert auch ein kanonischer, unendlicher k -Graph: Wir können jeden Knoten mit Beschriftung $x \not\leq_\mu k$ einfach stattdessen mit $\frac{1}{2}$ beschriften. Wir können jedoch erneut die Unentscheidbarkeit der Existenz eines endlichen k -Graphen zeigen.

Lemma 72

Sei μ ein Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$ eine endliche Menge. Die Existenz eines endlichen Graphen von X in μ ist unentscheidbar.

Beweis (Lemma 72)

wie im Beweis von Lemma 66, S. 54 sei Σ ein Alphabet und $\mathbb{K} = \Sigma^+ \cup \{\omega\}$ und $X = \Sigma^+$. Sei L eine kontextsensitive Sprache. Sei $a \in \Sigma$ und $X \subseteq \Sigma$. Für alle $w \in X^+$ sei $\mu(w) = a$, falls $w \in L$ und $\mu(w) = aa$, sonst. Sei $k = aa$ und \leq die natürliche Ordnung auf Σ^+ erweitert um $w < \omega$ für alle $w \in \Sigma^+$ und $\omega \leq \omega$. Sei G ein k -Graph von X in μ mit Wurzelknoten v_0 . Wir beobachten, dass kein Knoten mit Beschriftung $\frac{1}{2}$ existiert, da für alle $w \in \Sigma^+$ gilt: $\mu(w) \in \{a, aa\}$. Also ist G ein Graph von X in μ und wir können dieselbe Argumentation wie im Beweis von Lemma 66, S. 54 verwenden, um zu zeigen: Es existiert ein endlicher k -Graph von X in μ g.d.w. L regulär ist. Da die Regularität einer kontextsensitiven Sprache unentscheidbar ist, ist auch die Existenz eines endlichen k -Graphen von X in μ unentscheidbar.

Ist μ assoziativ und k eine Schwelle, können wir erneut einen kanonischen Graphen angeben:

Definition 73 (k -Graph eines assoziativen Kostenmodells)

Sei μ ein assoziatives Verbands-Kostenmodell über \mathbb{K} , k eine Schwelle von X in μ und $X \subseteq \mathbb{K}$. Seien $c, \frac{1}{2} \notin \mathbb{K}$. Wir definieren induktiv den Graphen $G_{\mu, X, k, c, \frac{1}{2}}$ mit Wurzelknoten c wie folgt:

- (1) Sei $x \in X$. Dann ist y mit

$$y = \begin{cases} \mu(x), & \text{falls } \mu(x) \leq_{\mu} k, \\ \frac{1}{2}, & \text{sonst,} \end{cases}$$

ein Knoten mit Beschriftung y und es gibt genau eine mit x beschriftete Kante von c nach y .

- (2) Sei $x \neq c, \frac{1}{2}$ ein Knoten und $x' \in X$. Sei $y = \mu(x\mu(x'))$.

a) Gelte $y \leq_{\mu} k$. Dann ist y ein Knoten mit Beschriftung y und es gibt genau eine mit x' beschriftete Kante von x nach y .

b) Gelte $y \not\leq_{\mu} k$. Dann ist $\frac{1}{2}$ ein Knoten mit Beschriftung $\frac{1}{2}$ und es gibt genau eine mit x' beschriftete Kante von x nach $\frac{1}{2}$.

- (3) Ist $\frac{1}{2}$ ein Knoten, gibt es für jedes $x \in X$ genau eine mit x beschriftete Kante von $\frac{1}{2}$ zu y mit

$$y = \begin{cases} \frac{1}{2}, & \text{falls } k \uparrow x, \\ z, & \text{falls } k \downarrow_z x. \end{cases}$$

Lemma 74

Sei μ ein assoziatives Verbands-Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Seien $c, \frac{1}{2} \notin \mathbb{K}$. Sei k eine Schwelle von X in μ .

Der Graph $G_{\mu, X, k, c, \frac{1}{2}}$ ist ein k -Graph von X in μ .

Beweis (Lemma 74)

Sei w ein Pfad von $G_{\mu, X, c}$, der im Wurzelknoten c startet, in x resultiert und die Kanten $e_1 \dots e_n$ mit jeweiligen Beschriftungen $x_1 \dots x_n$ traversiert. Sei $y = \mu(x_1 \dots x_n)$. Wir zeigen:

$$x = \begin{cases} y, & \text{falls } y \leq_{\mu} k \\ \frac{1}{2}, & \text{sonst.} \end{cases} \quad (41)$$

- (1) Sei $n = 1$.

a) Sei $y \leq_{\mu} k$. Dann folgt aus der ersten Regel: $x = y$.

b) Sei $y \not\leq_\mu k$. Dann folgt aus der ersten Regel: $x = \frac{1}{2}$.

(2) Gelte die Eigenschaft für w . Existiere eine mit x' beschriftete Kante von x zu einem Knoten z . Sei $y' = \mu(x_1 \dots x_n x')$. Aus der Assoziativität von μ folgt: $y' = \mu(x_1 \dots x_n x') = \mu(\mu(x_1 \dots x_n) \mu(x'))$. Wir zeigen:

$$z = \begin{cases} y', & \text{falls } y' \leq_\mu k \\ \frac{1}{2}, & \text{sonst} \end{cases} \quad (42)$$

a) Sei $x = y \neq \frac{1}{2}$. Dann gilt per Induktionsannahme $x = y \leq_\mu k$ und $y' = \mu(\mu(x_1 \dots x_n) \mu(x')) = \mu(x \mu(x'))$.

i. Sei $y' \leq_\mu k$. Dann gilt laut Regel 2a: $z = y'$.

ii. Sei $y' \not\leq_\mu k$. Dann gilt laut Regel 2b: $z = \frac{1}{2}$.

b) Sei $x = \frac{1}{2}$. Dann gilt per Induktionsannahme $y \not\leq_\mu k$.

i. Gelte $k \uparrow x'$. Dann gilt $y' \not\leq_\mu k$ und $z = \frac{1}{2}$.

ii. Gelte $k \downarrow_v x'$. Dann gilt $y' = v = z \leq_\mu k$.

Also ist $G_{\mu, X, k, c, \frac{1}{2}}$ ein k -Graph von X in μ .

Da jeder Knoten $v \neq c$ mit v beschriftet ist, gilt für zwei Knoten $v, v' \neq c$:

$$v \neq v' \text{ impliziert } \ell(v) \neq \ell(v') \quad (43)$$

Wie oben identifizieren wir alle Graphen $G_{\mu, X, k, c, \frac{1}{2}}$ für beliebige c und $\frac{1}{2}$ jedoch feste μ, X und k .

Notation 75 (k -Graph eines assoziativen Kostenmodells)

Sei μ ein assoziatives Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Sei k eine Schwelle von X in μ . Wir identifizieren alle Graphen $G_{\mu, X, k, c, \frac{1}{2}}$ mit $c, \frac{1}{2} \notin \mathbb{K}$ und schreiben $G_{\mu, X, k}$ anstatt $G_{\mu, X, c, \frac{1}{2}}$.

Analog zu Korollar 70, S. 56 gilt: Seien X und der k -Abschluss von X in einem assoziativen Kostenmodell μ endlich. Dann folgt aus Gleichung (43) direkt die Endlichkeit des kanonischen k -Graphen von X in μ :

Korollar 76

Sei μ ein assoziatives Kostenmodell über \mathbb{K} und $X \subseteq \mathbb{K}$. Sei k eine Schwelle von X in μ . Seien X und $(X)_{\leq k}^\mu$ endlich.

Dann ist $G_{\mu, X, k}$ endlich.

3.3.6 Kreuzprodukt

In dieser Arbeit betrachten wir aus formaler Sicht stets nur eine Art von Kosten mit einem Kostenmodell auf einmal. In diesem Abschnitt definieren wir einen Kompositionsoperator für Kostenmodelle mit dem Ziel trotzdem mehrere Arten von Kosten mit ihren jeweiligen Kostenmodellen betrachten zu können. Dabei folgt die Komposition intuitiv dem Kreuzprodukt: Ein Tupel $\langle a_1, \dots, a_n \rangle$ beschreibt Kosten von n verschiedenen Arten – das komponierte Kostenmodell kann Familien und Folgen solcher Tupel verrechnen.

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Wir definieren das *Kreuzprodukt*

$$\mu_1 \times \mu_2 := \langle \mathbb{K}, \sum : \text{FAM}(\mathbb{K}) \rightarrow \mathbb{K}, \prod : \mathbb{K}^+ \rightarrow \mathbb{K} \rangle \quad (44)$$

von μ_1 und μ_2 durch

$$\mathbb{K} := \mathbb{K}_1 \times \mathbb{K}_2 \quad (45)$$

$$\sum(f) := \langle \sum_1((x_i)_{i \in I}), \sum_2((y_i)_{i \in I}) \rangle \quad (46)$$

$$\prod(w) := \langle \prod_1(x_1 \dots x_n), \prod_2(y_1 \dots y_n) \rangle \quad (47)$$

für alle $f = (a_\ell)_{\ell \in L}$ mit $a_\ell = \langle x_i, y_i \rangle \in \mathbb{K}$ für alle $\ell \in L$ und

$$w = \langle x_1, y_1 \rangle \dots \langle x_n, y_n \rangle \in \mathbb{K}^+.$$

Dabei ist \sum wohldefiniert, da \sum_1 und \sum_2 jeweils totale Kompressoren sind. Wir zeigen, dass das Kreuzprodukt zweier Kostenmodelle wieder ein Kostenmodell ist.

Lemma 77

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Dann ist $\mu_1 \times \mu_2$ ein Kostenmodell.

Beweis (Lemma 77)

Zu zeigen: \sum ist index-unabhängig, identitätserhaltend und total. Jede der ersten beiden Eigenschaften lässt sich direkt aus der jeweiligen Eigenschaft von \sum_i ableiten. Die Totalität folgt direkt aus der Definition.

Wir betrachten im Folgenden die Abgeschlossenheit des Kreuzprodukts bezüglich der vorher in diesem Abschnitt eingeführten Unterklassen, Eigenschaften und Konstruktionen.

Sind zwei Kostenmodelle über vollständigen Verbänden definiert, ist ihr Kreuzprodukt auch über einem vollständigen Verband definiert.

Lemma 78

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell über einem vollständigen Verband $\langle \mathbb{K}_i, \leq_i \rangle$. Dann ist $\mu_1 \times \mu_2 = \langle \mathbb{K}, \sum, \prod \rangle$ ein Kostenmodell über dem vollständigen Verband $\langle \mathbb{K}_1, \leq_1 \rangle \times \langle \mathbb{K}_2, \leq_2 \rangle$.

Beweis (Lemma 78)

Wir zeigen, dass \sum jede Familie auf ihr Supremum bezüglich $\langle \mathbb{K}_1, \leq_1 \rangle \times \langle \mathbb{K}_2, \leq_2 \rangle$ abbildet. Per Definition gilt $\sum(f) = \langle \sum_1((x_i)_{i \in I}), \sum_2((y_i)_{i \in I}) \rangle$. Da μ_i über dem vollständigen Verband $\langle \mathbb{K}_i, \leq_i \rangle$ definiert ist, ist $\sum_1((x_i)_{i \in I})$ das Supremum z_1 von $(x_i)_{i \in I}$ in $\langle \mathbb{K}_1, \leq_1 \rangle$. Also ist $\sum(f) = \langle z_1, z_2 \rangle$ das Paar der Suprema z_i der Teilfamilien im entsprechenden Verband. Wie im Beweis von Lemma 3, S. 20 können wir zeigen, dass $\langle z_1, z_2 \rangle$ das Supremum von f ist.

Wir zeigen, dass der Abschluss des Kreuzprodukts eine Teilmenge des Kreuzprodukts der Abschlüsse ist. Die andere Richtung gilt im Allgemeinen nicht: Im Kreuzprodukt der Abschlüsse können Paare $\langle x, y \rangle$ vorkommen, wobei x und y nur mit unterschiedlich langen Sequenzen erzeugt werden können.

Lemma 79

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Sei $X_i \subseteq \mathbb{K}_i$. Dann gilt:

$$(X_1)^{\mu_1} \times (X_2)^{\mu_2} \supseteq (X_1 \times X_2)^{\mu_1 \times \mu_2}.$$

Beweis (Lemma 79)

Per Definition gilt: $(X_i)^{\mu_i} = \{\mu_i(w_i) \mid w_i \in X_i^+\}$.

$$\begin{aligned} (X_1)^{\mu_1} \times (X_2)^{\mu_2} &= \{\mu_1(w_1) \mid w_1 \in X_1^+\} \times \{\mu_2(w_2) \mid w_2 \in X_2^+\} \\ &= \{\langle \mu_1(w_1), \mu_2(w_2) \rangle \mid w_1 \in X_1^+, w_2 \in X_2^+\} \\ &= \{\langle \mu_1(w_1), \mu_2(w_2) \rangle \mid w_1 \in X_1^m, w_2 \in X_2^n, m, n \in \mathbb{N}_1\} \\ &\supseteq \{\langle \mu_1(w_1), \mu_2(w_2) \rangle \mid w_1 \in X_1^n, w_2 \in X_2^n, n \in \mathbb{N}_1\} \\ &= \{\langle \mu_1(x_1 \dots x_n), \mu_2(y_1 \dots y_n) \rangle \mid w = \langle x_1, y_1 \rangle \dots \langle x_n, y_n \rangle \in (X_1 \times X_2)^+\} \\ &= \{\mu(w) \mid w \in (X_1 \times X_2)^+\} = ((X_1 \times X_2))^\mu. \end{aligned}$$

Analog können wir das Lemma für den k -Abschluss zeigen:

Lemma 80

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Sei $k_i \in \mathbb{K}_i$ und $X_i \subseteq \mathbb{K}_i$. Dann gilt:

$$(X_1)_{\leq k_1}^{\mu_1} \times (X_2)_{\leq k_2}^{\mu_2} \supseteq (X_1 \times X_2)_{\leq \langle k_1, k_2 \rangle}^{\mu_1 \times \mu_2}.$$

Beweis (Lemma 80, S. 61)

Per Definition gilt: $(X_i)^{\mu_i}_{\leq k_i} = \{\mu_i(w_i) \mid w_i \in X_i^+, \mu_i(w_i) \leq_i k_i\}$.

$$\begin{aligned}
 & (X_1)^{\mu_1}_{\leq k_1} \times (X_2)^{\mu_2}_{\leq k_2} \\
 &= \{\mu_1(w_1) \mid w_1 \in X_1^+, \mu_1(w_1) \leq_1 k_1\} \times \{\mu_2(w_2) \mid w_2 \in X_2^+, \mu_2(w_2) \leq_2 k_2\} \\
 &= \{\langle \mu_1(w_1), \mu_2(w_2) \rangle \mid w_i \in X_i^+, \mu_i(w_i) \leq_i k_i, i = 1, 2\} \\
 &= \{\langle \mu_1(w_1), \mu_2(w_2) \rangle \mid w_i \in X_i^{m_i}, \mu_i(w_i) \leq_i k_i, i = 1, 2, m_i \in \mathbb{N}_1\} \\
 &\supseteq \{\langle \mu_1(w_1), \mu_2(w_2) \rangle \mid w_i \in X_i^n, \mu_i(w_i) \leq_i k_i, i = 1, 2\} \\
 &= \{\mu(w) \mid w \in (X_1 \times X_2)^+, \mu(w) \leq \langle k_1, k_2 \rangle\} = ((X_1 \times X_2))^{\mu}_{\leq \langle k_1, k_2 \rangle} .
 \end{aligned}$$

Wir zeigen, dass die folgenden Eigenschaften bei der Bildung des Kreuzprodukts erhalten bleiben: Kompositionalität, Assoziativität, Kommutativität.

Lemma 81

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Sei $X_i \subseteq \mathbb{K}_i$. Dann gilt für die Eigenschaften

$$e \in \{\text{Kompositionalität, Assoziativität, Kommutativität}\} :$$

Wenn μ_1 und μ_2 die Eigenschaft e erfüllen, erfüllt auch $\mu_1 \times \mu_2$ die Eigenschaft e .

Beweis (Lemma 81)

Sei $\mu_1 \times \mu_2 = \mu = \langle \mathbb{K}, \sum, \prod \rangle$.

- (1) Seien μ_1 und μ_2 kompositional. Zu zeigen: \sum ist kompositional. Seien I und J Indexmengen. Sei $f \in \text{FAM}_I(\mathbb{K})$ eine Familie und $g \in \text{FAM}_J(\text{FAM}(\mathbb{K}))$ eine Zerlegung von f in Teilfamilien. Zu zeigen: $\sum(f) = \sum((a_j)_{j \in J})$ mit $a_j = \sum(g(j))$.

Sei $f = (a_i)_{i \in I}$, $a_i = \langle x_i, y_i \rangle \in \mathbb{K}$ für alle $i \in I$. Sei $f_1 = (x_i)_{i \in I}$ und $f_2 = (y_i)_{i \in I}$.

Seien g_1 und g_2 die jeweiligen, g entsprechenden Zerlegungen von f_1 und f_2 , das heißt: $g(j)$ ist über Indexmenge L definiert g.d.w. $g_1(j)$ und $g_2(j)$ über Indexmenge L definiert sind, und $g(j)(\ell) = \langle g_1(j)(\ell), g_2(j)(\ell) \rangle$.

$$\sum(f) = \langle \sum_1(f_1), \sum_2(f_2) \rangle \quad (48)$$

$$= \langle \sum_1((b_j)_{j \in J}), \sum_2((c_j)_{j \in J}) \rangle , \quad (49)$$

mit $b_j = \sum_1(g_1(j))$ und $c_j = \sum_2(g_2(j))$ für alle $j \in J$.

Für alle $j \in J$ gilt: $a_j = \sum(g(j)) = \langle \sum_1(g_1(j)), \sum_2(g_2(j)) \rangle$. Also gilt: $a_j = \langle b_j, c_j \rangle$. Damit gilt $\sum((a_j)_{j \in J}) = \langle \sum_1((b_j)_{j \in J}), \sum_2((c_j)_{j \in J}) \rangle$.

- (2) Seien μ_1 und μ_2 assoziativ. Zu zeigen: μ ist assoziativ. Sei $u_1 = x_1 \dots x_m$, $u_2 = y_1 \dots y_m$, $v_1 = x'_1 \dots x'_n$ und $v_2 = y'_1 \dots y'_n$. Sei $u = \langle x_1, y_1 \rangle \dots \langle x_m, y_m \rangle$ und $v = \langle x'_1, y'_1 \rangle \dots \langle x'_n, y'_n \rangle$. Zu zeigen: $\Pi(uv) = \Pi(\Pi(u)\Pi(v))$.

$$\begin{aligned} \Pi(uv) &= \langle \Pi_1(u_1v_1), \Pi_2(u_2v_2) \rangle \\ &= \langle \Pi_1(\Pi_1(u_1)\Pi_1(v_1)), \Pi_2(\Pi_2(u_2)\Pi_2(v_2)) \rangle \\ &= \Pi(\langle \Pi_1(u_1)\Pi_2(u_2) \rangle \langle \Pi_1(v_1)\Pi_2(v_2) \rangle) \\ &= \Pi(\Pi(u)\Pi(v)) . \end{aligned}$$

- (3) Seien μ_1 und μ_2 kommutativ. Zu zeigen: μ ist kommutativ. Seien $v = \langle x_1, y_1 \rangle \dots \langle x_n, y_n \rangle$, $w = \langle x'_1, y'_1 \rangle \dots \langle x'_n, y'_n \rangle \in \mathbb{K}^+$ Permutationen. Zu zeigen: $\Pi(v) = \Pi(w)$.

$$\begin{aligned} \Pi(v) &= \langle \Pi_1(x_1 \dots x_n), \Pi_2(y_1 \dots y_n) \rangle \\ &= \langle \Pi_1(x'_1 \dots x'_n), \Pi_2(y'_1 \dots y'_n) \rangle \\ &= \Pi(w) . \end{aligned}$$

Wir zeigen, dass bestimmte Schwellen bei der Bildung des Kreuzprodukts erhalten bleiben. Intuitiv bleiben solche Schwellen erhalten, die in den einzelnen Kostenmodellen nie wieder unterschritten werden können.

Lemma 82

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \Pi_i \rangle$ ein Kostenmodell. Sei $X_i \subseteq \mathbb{K}_i$ und k_i eine Schwelle von X_i in μ_i mit $k_i \uparrow x$ für alle $x \in X_i$. Sei $X = X_1 \times X_2$.

Dann ist $\langle k_1, k_2 \rangle$ eine Schwelle von X in μ .

Beweis (Lemma 82)

Sei $\langle x_1, x_2 \rangle \in X$. Zu zeigen:

- (1) Entweder für alle $w \in X^+$ mit $\mu(w) \not\leq \langle k_1, k_2 \rangle$: $\mu(w \langle x_1, x_2 \rangle) \not\leq \langle k_1, k_2 \rangle$, oder
- (2) Es existiert ein $\langle y_1, y_2 \rangle \in X$, sodass für alle $w \in X^+$ mit $\mu(w) \not\leq \langle k_1, k_2 \rangle$: $\mu(w \langle x_1, x_2 \rangle) = \langle y_1, y_2 \rangle$.

Sei $w \in X^+$ mit $\mu(w) = \langle z_1, z_2 \rangle \not\leq \langle k_1, k_2 \rangle$. Sei $w = \langle u_1, v_1 \rangle \dots \langle u_n, v_n \rangle$,

$u = u_1 \dots u_n$ und $v = v_1 \dots v_n$. Wir zeigen: $\mu(w\langle x_1, x_2 \rangle) \not\leq \langle k_1, k_2 \rangle$.

$$\begin{aligned}\mu(w\langle x_1, x_2 \rangle) &= \prod(w\langle x_1, x_2 \rangle) \\ &= \langle \prod_1(ux_1), \prod_2(vx_2) \rangle\end{aligned}$$

Da k_i eine Schwelle von X_i in μ_i ist mit $k_i \uparrow x_i$, gelten: $\prod_1(ux_1) \not\leq_1 k_1$ und $\prod_2(vx_2) \not\leq_2 k_2$.

Wir zeigen, dass 2. nicht gilt. Zu zeigen: Es existieren $w \neq w' \in X^+$ mit $\mu(w) \not\leq \langle k_1, k_2 \rangle$, $\mu(w') \not\leq \langle k_1, k_2 \rangle$ und $\mu(w\langle x_1, x_2 \rangle) \neq \mu(w'\langle x_1, x_2 \rangle)$. Sei $w = \langle u_1, v_1 \rangle \dots \langle u_n, v_n \rangle$, $u = u_1 \dots u_n$, $v = v_1 \dots v_n$.

$$\mu(w\langle x_1, x_2 \rangle) = \langle \prod_1(ux_1), \prod_2(vx_2) \rangle.$$

Da k_i eine Schwelle von X_i in μ_i ist mit $k_i \uparrow x_i$, existieren u' und v' , sodass $\prod_1(ux_1) \neq \prod_1(u'x_1)$ und $\prod_2(vx_2) \neq \prod_2(v'x_2)$. Sei $u' = u'_1 \dots u'_n$ und $v' = v'_1 \dots v'_n$. Sei $w' = \langle u'_1, v'_1 \rangle \dots \langle u'_n, v'_n \rangle$. Dann gilt:

$$\mu(w'\langle x_1, x_2 \rangle) = \langle \prod_1(ux'_1), \prod_2(vx'_2) \rangle.$$

Für Schwellen, die den konstanten Rücksprung erlauben, können wir die Erhaltung nicht zeigen. Nehmen wir beispielsweise das Kostenmodell der Max-Plus-Algebra für μ_1 und μ_2 an: Wählen wir X_1, X_2 mit $-\infty, 0, 1 \in X_1, X_2$ und $\omega \notin X_1 \cup X_2$, dann sind natürliche Zahlen $k > 0$ jeweils Schranken von X_1 und X_2 . Jedoch ist $\langle k, k \rangle$ keine Schranke von $X_1 \times X_2$: Die Kosten eines Wortes $v \in \langle \langle 1, 1 \rangle \rangle^n$ mit $n > k$ überschreiten k . Gleiches gilt für die Kosten des Wortes $w \in \langle \langle 1, 0 \rangle \rangle^n$ mit $n > k$. Für das Paar $x = \langle -\infty, 1 \rangle$ gilt: $y_v = \mu(vx) \not\leq k$, $y_w = \mu(wx) \leq k$ und $y_v \neq y_w$. Daher können wir in diesem Fall nur das folgende, abgeschwächte Lemma zeigen:

Lemma 83

Sei für $i \neq j \in \{1, 2\}$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Sei $y_i \in \mathbb{K}_i$. Sei $X_i \subseteq \mathbb{K}_i$ und k_i eine Schwelle von X_i in μ_i mit $k_i \downarrow_{y_i} x$ für alle $x \in X_i$ und $y_i \leq_i k_i$. Sei $X \subseteq X_1 \times X_2$, sodass für alle $\langle x_1, x_2 \rangle \in X$ gilt:

- (1) $k_i \downarrow_{z_i} x_i$ impliziert:
 - a) $k_j \downarrow_{z_j} x_j$
 - b) $z_i = y_i$, und
 - c) $z_j = y_j$, und
 - d) für alle $w \in X_i^+$ gilt: $\mu_i(wx_i) = y_i$,
 - e) für alle $w \in X_j^+$ gilt: $\mu_j(wx_j) = y_j$.

Dann ist $\langle k_1, k_2 \rangle$ eine Schwelle von X in μ .

Beweis (Lemma 83, S. 64)

Sei $\langle x_1, x_2 \rangle \in X$. Zu zeigen:

- (1) Entweder für alle $w \in X^+$ mit $\mu(w) \not\leq \langle k_1, k_2 \rangle$: $\mu(w\langle x_1, x_2 \rangle) \not\leq \langle k_1, k_2 \rangle$, oder
 - (2) Es existiert ein $\langle y_1, y_2 \rangle \in X$, sodass für alle $w \in X^+$ mit $\mu(w) \not\leq \langle k_1, k_2 \rangle$: $\mu(w\langle x_1, x_2 \rangle) = \langle y_1, y_2 \rangle$.
- (1) Gelte $k_i \uparrow x_i$. Dann gilt per Voraussetzung $k_j \uparrow x_j$. Analog zum Beweis von Lemma 82, S. 63 können wir zeigen, dass immer Fall 1 eintritt.
 - (2) Gelte $k_i \downarrow_{z_i} x_i$. Dann gilt $k_j \downarrow_{z_j} x_j$, $z_i = y_i$, $z_j = y_j$, für alle $u \in X_i^+$ gilt: $\mu_i(ux_i) = y_i$, und für alle $v \in X_j^+$ gilt: $\mu_j(vx_j) = y_j$. Also tritt für alle Worte $w \in X^+$ Fall 2 ein: $\mu(w\langle x_1, x_2 \rangle) = \langle y_1, y_2 \rangle \leq \langle k_1, k_2 \rangle$.

Wir definieren das Kreuzprodukt auf Graphen von Kostenmodellen. Sei G_i ein Graph einer Menge X_i in einem Kostenmodell μ_i mit Wurzelknoten v_i . Dann definieren wir den Graphen $G_1 \times G_2$ wie folgt:

- (1) $\langle v_1, v_2 \rangle$ ist der Wurzelknoten,
- (2) Sei $\langle x_1, x_2 \rangle$ ein Knoten von $G_1 \times G_2$. Existiere in G_i eine Kante mit Beschriftung y_i von x_i zu einem Knoten z_i mit Beschriftung z'_i . Dann ist $\langle z_1, z_2 \rangle$ ein mit $\langle z'_1, z'_2 \rangle$ beschrifteter Knoten von $G_1 \times G_2$ und es existiert in $G_1 \times G_2$ eine mit $\langle y_1, y_2 \rangle$ beschriftete Kante von $\langle x_1, x_2 \rangle$ zu $\langle z_1, z_2 \rangle$.

Wir zeigen, dass $G_1 \times G_2$ ein Graph von $X_1 \times X_2$ in $\mu_1 \times \mu_2$ ist:

Lemma 84

Sei für $i = 1, 2$, $\mu_i = \langle \mathbb{K}_i, \sum_i, \prod_i \rangle$ ein Kostenmodell. Sei $X_i \subseteq \mathbb{K}_i$ eine nicht-leere Menge und G_i ein Graph von X_i in μ_i .

$G_1 \times G_2$ ist ein Graph von $X_1 \times X_2$ in $\mu_1 \times \mu_2$.

Beweis (Lemma 84)

Sei $\mu = \mu_1 \times \mu_2$. Zu zeigen: Es (1) existiert genau ein Knoten v_0 ohne eingehende Kanten und (2) alle Knoten sind von v_0 aus erreichbar. (3) Es existiert ein $\langle c_1, c_2 \rangle \notin \mathbb{K}_1 \times \mathbb{K}_2$ mit: $\ell: V \cup E \rightarrow \mathbb{K} \cup \{\langle c_1, c_2 \rangle\}$ ist die Beschriftungsfunktion. (4) Jeder Knoten hat für jedes $x \in X_1 \times X_2$ eine mit x beschriftete, ausgehende Kante. Es gilt (5) $\ell(v_0) = \langle c_1, c_2 \rangle$ und (6) für alle von v_0 ausgehenden und in einem Knoten v resultierende Pfade

über sukzessive Kanten $e_1 \dots e_n$ gilt:

$$e_1 \dots e_n \in (X_1 \times X_2)^+ \quad (50)$$

$$\mu(\ell(e_1) \dots \ell(e_n)) = \ell(v) \quad (51)$$

Sei v_i der jeweilige Wurzelknoten von G_i und c_i die Beschriftung von v_i . Per Definition ist $\langle v_1, v_2 \rangle$ ein Knoten von $G_1 \times G_2$. Da v_i in G_i keine eingehenden Kanten hat, jedoch alle anderen Knoten von G_i eingehende Kanten haben und von v_i erreichbar sind, erfüllt $v = \langle v_1, v_2 \rangle$ die Eigenschaft (1). (2) folgt daraus, dass induktiv nur von v_0 erreichbare Knoten eingefügt werden. Sei ℓ_i die jeweilige Beschriftungsfunktion. Wir beobachten: ℓ_i bildet jeden Knoten auf ein Element aus $\mathbb{K} \cup \{c_i\}$ ab. Dann folgen (3), (4) und (5) direkt, da die Beschriftungen übernommen werden. (6) kann man leicht per Induktion über den Aufbau der Pfade zeigen.

Analog können wir auch das Kreuzprodukt zweier k_i -Graphen G_i von X_i in μ_i bilden. Hier identifizieren wir jedoch alle Knoten, in deren Beschriftung $\frac{1}{2}$ vorkommt. Ist $\langle k_1, k_2 \rangle$ eine Schwelle von $X_1 \times X_2$, ist $G_1 \times G_2$ auch ein $\langle k_1, k_2 \rangle$ -Graph von $X_1 \times X_2$ in $\mu_1 \times \mu_2$.

Würden wir anstatt der induktiven Definition von $G_1 \times G_2$ direkt das Kreuzprodukt der Knoten und Kanten wählen, würden neue Zustände dazu kommen – aus demselben Grund aus dem die Rückrichtung von Lemma 79, S. 61 nicht gilt.

3.4 Semantik gewichteter Netze

Wir definieren zwei Semantiken für gewichtete Netze:

- (1) Die *Wert-Semantik* beschreibt die Kosten eines gewichteten Netzes in einem einzigen, aggregierten Wert.
- (2) Die *Sprach-Semantik* beschreibt die Kosten eines beschrifteten, gewichteten Netzes als eine *gewichtete Sprache*.

Der *Wert* eines gewichteten Netzes ergibt sich aus einer Perspektive und einem Kostenmodell in den folgenden Schritten:

- (1) Wende die Perspektive auf die Menge der Abläufe an.
- (2) Wende die Kostenfunktion auf jedes resultierende Wort an. Das Resultat ist eine Familie von Wörtern über Kosten.
- (3) Wende das Kostenmodell auf die resultierende Familie an.

Definition 85 (Wert eines gewichteten Netzes)

Sei N ein gewichtetes Netz über \mathbb{K} . Sei μ ein Kostenmodell über \mathbb{K} . Sei π eine Perspektive von N .

Dann ist $\llbracket N \rrbracket_{\mu}^{\pi} := \mu(\llbracket \pi(N) \rrbracket_N)$ der Wert von N bezüglich π und μ .

Beispiel 86

Wir betrachten die Wert-Semantik der Netze Net_{13} aus Abbildung 13, S. 44, Net_{15} aus Abbildung 15, Net_{17} aus Abbildung 17 und Net_{16} aus Abbildung 16 unter dem Kostenmodell der Max-Plus-Algebra und unterschiedlichen Perspektiven.

Sei in jedem der Netze die jeweilige Anfangsmarkierung die einzige Zielmarkierung und die Menge der Rahmenmarkierungen uneingeschränkt.

Sei $\Sigma = \{b, n, r\}$ und U der Anwendungsfall mit $\bullet U = \Sigma^* \{b\}$ und $U^{\bullet} = \{n, r\}$. Sei für $N \in \{\text{Net}_{13}, \text{Net}_{15}, \text{Net}_{17}, \text{Net}_{16}\}$ jeweils N' die Variante, in der die Transitionen die folgende Beschriftung aufweisen: $t_1 \mapsto b$, $t_2 \mapsto n$ und $t_3 \mapsto r$ und $t \mapsto \varepsilon$ für die übrigen Transitionen.

Netz	Perspektive	Wert
Net_{13}	$\text{FILTER}[\{t_1, t_2, t_3\}]$	ω
Net_{13}	$\text{FILTER}[\{t_1, t_3\}]$	0
Net_{15}	$\text{FILTER}[\{t_1, t_2, t_3\}]$	0
Net_{17}	$\text{FILTER}[\{t_1, t_2, t_3\}]$	ω
Net_{16}	$\text{FILTER}[\{t_1, t_2, t_3\}]$	k
Net_{13}	$\text{FILTER}[\{t_1, t_2, t_3\}] \triangleright \text{AF}[\text{Net}_{13}, U]$	1
Net_{15}	$\text{FILTER}[\{t_1, t_2, t_3\}] \triangleright \text{AF}[\text{Net}_{15}, U]$	0
Net_{17}	$\text{FILTER}[\{t_1, t_2, t_3\}] \triangleright \text{AF}[\text{Net}_{17}, U]$	1
$\text{Net}_{16} (k = 0)$	$\text{FILTER}[\{t_1, t_2, t_3\}] \triangleright \text{AF}[\text{Net}_{16}, U]$	0
$\text{Net}_{16} (k > 0)$	$\text{FILTER}[\{t_1, t_2, t_3\}] \triangleright \text{AF}[\text{Net}_{16}, U]$	1

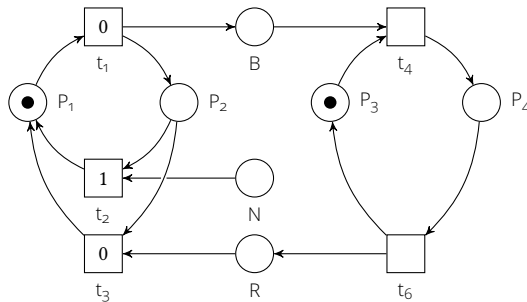


Abbildung 15 Ein gewichtetes Netz Net_{15}

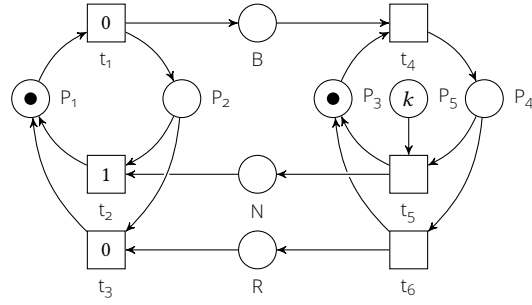


Abbildung 16 Ein Schema Net_{t_6} eines gewichteten Netzes

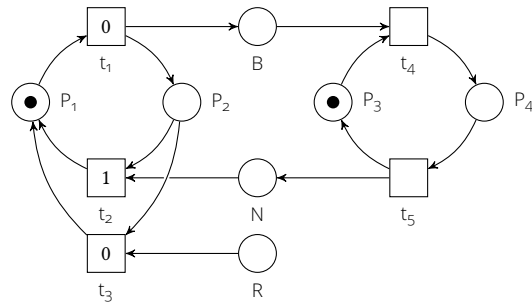


Abbildung 17 Ein gewichtetes Netz Net_{t_7}

Für ein beschriftetes, gewichtetes Netz definieren wir zusätzlich seine *gewichtete Sprache*.

Definition 87 (Sprache eines gewichteten Netzes)

Sei N ein beschriftetes, gewichtetes Netz mit Beschriftungsfunktion ℓ . Sei π eine Perspektive von G und μ ein Kostenmodell über \mathbb{K} . Sei $\pi_w = \text{WFILTER}[N, w] \triangleright \pi$.

Wir definieren die *Sprache* $\|N\|_{\pi, \mu}: \Sigma^* \rightarrow \mathbb{K}$ von N bezüglich π und μ durch

$$w \mapsto \llbracket N \rrbracket_{\mu}^{\pi_w} . \quad (52)$$

Falls jede Transition in N mit ε beschriftet ist, fallen der Wert des Wortes ε in der Sprache von N mit dem Wert von N zusammen.

Lemma 88

Sei N ein beschriftetes, gewichtetes Netz mit Transitionen T und Beschriftungsfunktion ℓ , sodass für alle Transitionen $t \in T$ gilt:

$$\ell(t) = \varepsilon . \quad (53)$$

Sei π eine Perspektive von N und μ ein Kostenmodell über \mathbb{K} . Dann gilt

$$\llbracket N \rrbracket_{\mu}^{\pi} = \llbracket N \rrbracket_{\pi, \mu}(\varepsilon) . \quad (54)$$

Beweis (Lemma 88)

$$\llbracket N \rrbracket_{\pi, \mu}(\varepsilon) \stackrel{\text{Def. 87, S. 68}}{=} \llbracket N \rrbracket_{\mu}^{\pi_{\varepsilon}} \quad (55)$$

$$\stackrel{\text{Def. 85, S. 67}}{=} \mu((\llbracket w \rrbracket_N)_{w \in \pi_{\varepsilon}(\text{Runs}(N))}) \quad (56)$$

$$\stackrel{\text{Gleichung (53)}}{=} \mu((\llbracket w \rrbracket_N)_{w \in \pi_{\varepsilon}(\text{Runs}(N, \varepsilon))}) \quad (57)$$

$$\stackrel{\text{Gleichung (24), S. 43}}{=} \mu((\llbracket w \rrbracket_N)_{w \in \pi(\text{Runs}(N, \varepsilon))}) \quad (58)$$

$$\stackrel{\text{Not. 43, S. 43}}{=} \mu((\llbracket w \rrbracket_N)_{w \in \pi(N)}) \quad (59)$$

$$\stackrel{\text{Def. 87, S. 68}}{=} \llbracket N \rrbracket_{\mu}^{\pi} . \quad (60)$$

Das Kostenmodell und die Perspektive sind die wesentlichen Parameter der Semantik gewichteter Netze. Einige Perspektiven hängen dabei eng zusammen, insbesondere dann, wenn das Kostenmodell über einem vollständigen Verband definiert ist: Liefert eine Perspektive π stets eine Teilmenge einer anderen Perspektive π' , ist der Wert von N bezüglich π höchstens der Wert von N bezüglich π' .

Lemma 89

Sei N ein gewichtetes Netz über \mathbb{K} . Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Seien π, π' Perspektiven, sodass $\pi(N) = (v_i)_{i \in I}$, $\pi'(N) = (w_j)_{j \in J}$, $I \subseteq J$ und $v_i = w_i$ für alle $i \in I$.

Dann gilt $\llbracket N \rrbracket_{\mu}^{\pi} \leq_{\mu} \llbracket N \rrbracket_{\mu}^{\pi'}$.

Beweis (Lemma 89)

Da μ über dem vollständigen Verband $\langle \mathbb{K}, \leq_{\mu} \rangle$ definiert ist, entsprechen $\llbracket N \rrbracket_{\mu}^{\pi} = \mu(\pi(N))$ und $\llbracket N \rrbracket_{\mu}^{\pi'} = \mu(\pi'(N))$ den jeweiligen Suprema der Familien. Aus $I \subseteq J$ und $v_i = w_i$ f.a. $i \in I$, schließen wir $\llbracket N \rrbracket_{\mu}^{\pi} \leq_{\mu} \llbracket N \rrbracket_{\mu}^{\pi'}$.

Dies gilt beispielsweise für die Perspektiven ID und $TERM[N]$, da jeder terminierende Ablauf ein Ablauf ist. Dies lässt sich weiter auf Filter-Perspektiven fortsetzen.

Korollar 90

Sei N ein Netz. Sei T eine Menge von Transitionen von N . Sei $\pi = \text{FILTER}[T] \triangleright \text{TERM}[N]$ und $\pi' = \text{FILTER}[T]$. Sei μ ein Verbands-Kostenmodell über \mathbb{K} .

Dann gilt $\llbracket N \rrbracket_{\mu}^{\pi} \leq_{\mu} \llbracket N \rrbracket_{\mu}^{\pi'}$.

3.5 Kosteneindeutigkeit

In diesem Abschnitt betrachten wir die für uns zentrale Eigenschaft der *Kosteneindeutigkeit* eines gewichteten Netzes. Abstrakt fordert Kosteneindeutigkeit, dass die Kosten aller Abläufe, die in der gleichen Markierung resultieren, gleich sind. In dieser Arbeit spielen kosteneindeutige, gewichtete Netze eine besondere Rolle in der Reduktion neuer auf gelöste Analysefragen.

Dazu definieren wir zunächst die Kosten einer Markierung m durch Anwenden von Perspektive und Kostenmodell auf die Menge der Abläufe, die in m resultieren.

Notation 91 (Kosten einer Markierung)

Sei N ein gewichtetes Netz über \mathbb{K} . Sei μ ein Kostenmodell über \mathbb{K} . Sei π eine Perspektive von N . Wir schreiben $\llbracket N, m \rrbracket_{\mu}^{\pi}$ für den Wert

$$\mu(\llbracket \pi(\text{RUNS}(N, m) \setminus \{\varepsilon\}) \rrbracket_N) .$$

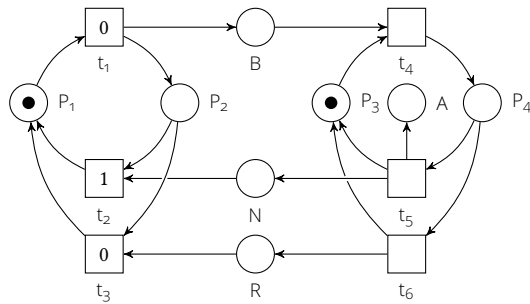


Abbildung 18 Ein gewichtetes Netz Net_{18}

Beispiel 92

Wir betrachten die Netze Net_{13} aus Abbildung 13, S. 44 und Net_{18} aus Abbildung 18. In der folgenden Tabelle betrachten wir das Kostenmodell

der Max-Plus-Algebra sowie die Perspektive $\text{FILTER}[t_1, t_2, t_3]$:

Markierung m	$\llbracket \text{Net}_{13}, m \rrbracket_{\mu}^{\pi}$	$\llbracket \text{Net}_{18}, m \rrbracket_{\mu}^{\pi}$
$\{P_1, P_3\}$	ω	0
$\{P_2, B, P_3\}$	ω	0
$\{P_2, P_4\}$	ω	0
$\{P_2, N, P_3\}$	ω	Nicht erreichbar
$\{P_2, N, A, P_3\}$	Nicht erreichbar	0
$\{P_1, P_3\}$	ω	Nicht erreichbar
$\{P_1, A, P_3\}$	Nicht erreichbar	1
$\{P_1, A/k, P_3\}$	Nicht erreichbar	k

Sind Perspektive und Kostenmodell kompositional, können wir aus den Kosten der erreichbaren Markierungen die Wert-Semantik des Netzes ableiten:

Satz 93 (Kompositionalität und Kosten von Markierungen)

Sei N ein gewichtetes Netz über \mathbb{K} . Sei μ ein kompositionales Kostenmodell über \mathbb{K} . Sei π eine kompositionale Perspektive von N . Dann gilt

$$\llbracket N \rrbracket_{\mu}^{\pi} = \mu((\llbracket N, m \rrbracket_{\mu}^{\pi})_{m \in \text{REACH}(N)}) \quad . \quad (61)$$

Beweis (Satz 93)

$\llbracket N, m \rrbracket_{\mu}^{\pi} = \mu((\llbracket w \rrbracket_N)_{w \in \pi(\text{RUNS}(N, m))})$, also gilt

$$\mu((\llbracket N, m \rrbracket_{\mu}^{\pi})_{m \in \text{REACH}(N)}) \quad (62)$$

$$= \mu((\mu((\llbracket w \rrbracket_N)_{w \in \pi(\text{RUNS}(N, m))}))_{m \in \text{REACH}(N)}) \quad (63)$$

$$= \sum((\mu((\llbracket w \rrbracket_N)_{w \in \pi(\text{RUNS}(N, m))}))_{m \in \text{REACH}(N)}) \quad (64)$$

$$= \sum((\sum((\prod(\llbracket w \rrbracket_N))_{w \in \pi(\text{RUNS}(N, m))}))_{m \in \text{REACH}(N)}) \quad (65)$$

Da μ kompositional ist, gilt

$$\sum((\sum((\prod(\llbracket w \rrbracket_N))_{w \in \pi(\text{RUNS}(N, m))}))_{m \in \text{REACH}(N)}) \quad (66)$$

$$= \sum((\prod(\llbracket w \rrbracket_N))_{w \in \pi(\text{RUNS}(N, m))}, m \in \text{REACH}(N)) \quad (67)$$

Da auch π kompositional ist, gilt

$$\sum((\prod(\llbracket w \rrbracket_N))_{w \in \pi(\text{RUNS}(N, m))}, m \in \text{REACH}(N)) \quad (68)$$

$$= \sum((\prod(\llbracket w \rrbracket_N))_{w \in \pi(N)}) \quad (69)$$

$$= \mu((\llbracket w \rrbracket_N)_{w \in \pi(N)}) \quad (70)$$

$$= \llbracket N \rrbracket_{\mu}^{\pi} \quad . \quad (71)$$

Sind Perspektive und Kostenmodell nicht kompositional, ist es möglich, dass die Wert-Semantik sich nicht aus den Kosten der erreichbaren Markierungen ableiten lässt.

Beispiel 94

Sei beispielsweise das Kostenmodell μ über den durch ∞ erweiterten reellen Zahlen so gewählt, dass jede endlichen Menge von Werten auf ihren Durchschnitt und jede unendliche Menge von Werten auf ihr Supremum abgebildet wird. Sei nun N ein beschränktes, gewichtetes Netz über reellen Zahlen und enthalte N einen Zyklus mit Kosten 0. Dann hat N unendlich viele Abläufe, die Wert-Semantik wählt den teuersten aus. Bilden wir jedoch zunächst die Kosten jeder erreichbaren Markierung erhalten wir nur endlich viele Werte, aus denen wir anschließend den Durchschnitt bilden, wobei zwangsläufig ein anderer Wert entsteht.

Daher betrachten wir im Rest dieses Abschnitts ausschließlich kompositionale Perspektiven und Kostenmodelle.

Es existieren gewichtete Netze N , sodass die Kosten jedes Ablaufs w , der in einer Markierung m resultiert, mit den Kosten der resultierenden Markierung m übereinstimmen. Dann nennen wir N *kosteneindeutig*. Formal schwächen wir den Begriff leicht ab, indem wir die Kosteneindeutigkeit durch eine Menge \mathbb{K}' von Kostenwerten parametrisieren: N ist \mathbb{K}' -*kosteneindeutig*, falls alle Markierungen mit Kosten aus \mathbb{K}' die Eigenschaft erfüllen.

Definition 95 (Kosteneindeutigkeit)

Sei N ein gewichtetes Netz über \mathbb{K} mit Plätzen P . Sei $P' \subseteq P$ und $\mathbb{K}' \subseteq \mathbb{K}$. Sei μ ein kompositionales Kostenmodell über \mathbb{K} . Sei π eine kompositionale Perspektive von N . Gelte für alle Markierungen m, m' von N mit $m|_{P'} = m'|_{P'}$ und alle Abläufe $w \in \text{RUNS}(N, m)$, $w' \in \text{RUNS}(N, m')$: Falls $\llbracket \pi(w) \rrbracket_N^\pi \in \mathbb{K}'$, dann $\mu(\llbracket \pi(w) \rrbracket_N) = \mu(\llbracket \pi(w') \rrbracket_N) = \llbracket N, m \rrbracket_\mu^\pi = \llbracket N, m' \rrbracket_\mu^\pi = \mu(\{\llbracket N, m \rrbracket_\mu^\pi, \llbracket N, m' \rrbracket_\mu^\pi\})$.

Dann ist N ein P', \mathbb{K}' -*kosteneindeutiges* Netz bzgl. π und μ .

Notation 96 (Kosteneindeutigkeit)

Sei N ein gewichtetes Netz mit Plätzen P über \mathbb{K} . Sei N P', \mathbb{K}' -kosteneindeutig. Falls $P = P'$ ($\mathbb{K} = \mathbb{K}'$) lassen wir den Parameter P' (\mathbb{K}') weg. Falls N beispielsweise P, \mathbb{K} -kosteneindeutig ist, nennen wir N nur *kosteneindeutig*.

Wir merken an, dass die Wahl der Perspektive und des Kostenmodells eine entscheidende Rolle spielt. Es ist durchaus möglich, dass N kosteneindeutig

für eine jedoch nicht eine andere Kombination aus Kostenmodell und Perspektive ist. Wir können jedoch beobachten, dass Netze mit einem baumförmigen Zustandsraum stets kosteneindeutig sind.

Beispiel 97

Wir betrachten die Netze Net_{13} aus Abbildung 13, S. 44 und Net_{18} aus Abbildung 18, S. 70. Wir beobachten, dass Net_{13} nicht kosteneindeutig ist: Der Ablauf $t_1 t_4 t_5 t_2$ hat Kosten 1 und resultiert in der Anfangsmarkierung. Die Anfangsmarkierung hat jedoch Kosten ω . Dagegen ist das Netz Net_{18} kosteneindeutig, da die Kosten einer Markierung stets mit der Anzahl der Marken auf A korrelieren: Ist N in einer erreichbaren Markierung m markiert, hat m die Kosten $m(A) - 1$ und ansonsten $m(A)$.

In den folgenden beiden Unterabschnitten betrachten wir das Problem, zu einem gegebenen gewichteten Netz ein kostenäquivalentes, kosteneindeutiges gewichtetes Netz zu bilden.

3.5.1 Kosteneindeutige äquivalente Netze

Sei N ein gewichtetes Netz über \mathbb{K} . Sei π eine kompositionale Perspektive von N und μ ein kompositionales Kostenmodell über \mathbb{K} . Wir zeigen die Existenz eines Netzes N' , sodass gilt:

- (1) $\llbracket N \rrbracket_{\mu}^{\pi} = \llbracket N' \rrbracket_{\mu}^{\pi}$,
- (2) N' ist kosteneindeutig.

Wir zeigen die Unentscheidbarkeit der Endlichkeit des konstruierten Netzes.

Anschließend präsentieren wir alternative Konstruktionsvorschriften für Teilklassen von Perspektiven und Kostenmodellen und untersuchen, wann diese zu einem endlichen Ergebnis führen.

3.5.1.1 Allgemeine Konstruktion

Dazu definieren wir zunächst die folgende Relation auf Abläufen: Zwei Abläufe $v, w \in \text{Runs}(N, m)$ seien äquivalent, geschrieben $v \sim w$ g.d.w.

- (1) $\mu(\llbracket \pi(\mathbf{v}) \rrbracket_N) = \mu(\llbracket \pi(\mathbf{w}) \rrbracket_N) = \mu(\pi(\llbracket u \rrbracket_N)_{u \in \{v, w\}})$, oder
- (2) $v = w$.

Lemma 98

\sim ist eine Äquivalenzrelation.

Beweis (Lemma 98, S. 73)

Reflexivität können wir direkt aus der Definition ablesen. Symmetrie und Transitivität ergeben sich jeweils aus der Symmetrie der Gleichheit.

Sei N' das kleinste Netz, das die folgenden Eigenschaften erfüllt:

- (1) Sei $w \in \text{RUNS}(N, m)$. Dann ist $[w]_{\sim}$ ein Platz von N' .
- (2) Sei $w \in \text{RUNS}(N, m)$ und t eine Transition. Dann ist $\langle [w]_{\sim}, t, [wt]_{\sim} \rangle$ eine Transition von N' mit Kosten $\llbracket t \rrbracket_N$, die eine Marke von $[w]_{\sim}$ konsumiert und eine Marke auf $[wt]_{\sim}$ produziert.
- (3) ε ist der einzige initial markierte Platz von N' .
- (4) Sei t eine Transition von N mit $t \in \text{RUNS}(N)$. Dann ist t eine Transition von N' , die eine Marke von ε konsumiert und eine Marke auf $[t]_{\sim}$ produziert.

In N' werden also erreichbare Markierungen entsprechend der Kosten aufgespalten. Aus der Kompositionalität von Kostenmodell und Perspektive können wir folgern:

Lemma 99

$$\llbracket N \rrbracket_{\mu}^{\pi} = \llbracket N' \rrbracket_{\mu}^{\pi}$$

Da wir die Markierungen entsprechend \sim aufsplitten, können wir zeigen:

Lemma 100

N' ist kosteneindeutig.

Korollar 101

Zu jedem Netz N existiert ein Netz N' mit

- (1) $\llbracket N \rrbracket_{\pi}^{\mu} = \llbracket N' \rrbracket_{\pi}^{\mu}$, und
- (2) N' ist kosteneindeutig.

Wir beobachten, dass N' für unbeschränkte N unendlich ist, da \sim per Definition in unendlich viele Äquivalenzklassen zerfällt. Wir zeigen im Folgenden die Unentscheidbarkeit der Endlichkeit von N' für beschränkte N . Dazu reduzieren wir das Endlichkeitsproblem für kontext-sensitive Sprachen (gegeben als *linear-beschränkte Turingmaschine*) auf die Endlichkeit von N' .

Satz 102 (Unentscheidbarkeit der Endlichkeit der Konstruktion)

Die Endlichkeit von N' ist für beschränkte N unentscheidbar.

Beweis (Satz 102, S. 74)

N' ist offenbar endlich, falls die Menge $\text{RUNS}(N)$ in endlich viele Äquivalenzklassen bezüglich \sim zerfällt. Sei L eine kontext-sensitive Sprache über einem Alphabet Σ , gegeben als linear-beschränkte Turingmaschine M . Sei $\perp \notin \Sigma$ ein Symbol. Sei $\mathbb{K} = (\Sigma \cup \{\perp\})^*$. Sei μ so gewählt, dass gilt: Sei $w \in \mathbb{K}^+$. Dann gilt $\mu(w) = w$, falls $w \in L$ und $\mu(w) = \perp$, sonst.

Sei N so gewählt, dass gilt:

- (1) Σ ist die Transitionsmenge von N .
- (2) $\llbracket \cdot \rrbracket_N$ ist die Identität.
- (3) $\Sigma^+ = \text{RUNS}(N)$.

Dann existiert eine Bijektion auf den Äquivalenzklassen auf $\text{RUNS}(N)$ bezüglich \sim und $L \setminus \{\varepsilon\} \cup \perp$. Also fällt die Endlichkeit der Menge der Äquivalenzklassen mit der Endlichkeit von L zusammen. Die Endlichkeit von L ist unentscheidbar. Also ist auch die Endlichkeit von N' unentscheidbar.

3.5.1.2 Alternative Konstruktion für Filter-Perspektiven und Verbände

Wir betrachten eine Filter-Perspektive $\pi = \text{FILTER}[T']$ und ein Verbands-Kostenmodell μ über \mathbb{K} . Sei \underline{G} das Netz, der aus einem Graphen G von $\mathbb{K}[N, \pi]$ in μ mit Wurzelzustand r entsteht. Dabei ist $\wr r \wr$ die Anfangsmarkierung von \underline{G} .

Wir konstruieren eine kosteneindeutige Variante $\text{REP}_\mu^\pi(N, G)$ von N wie folgt: Sei $\text{REP}_\mu^\pi(N, G)$ das synchrone Produkt aus N und \underline{G} , wobei wir eine Transition $t \in T'$ genau dann mit einer Transition e von \underline{G} synchronisieren, wenn e einem $\llbracket t \rrbracket_N$ -Übergang in G entspricht (siehe Abschnitt 2.7.7, S. 35). Das Verfahren ist abstrakt in Abbildung 19 dargestellt.

Lemma 103

Sei Q die Menge der Plätze von \underline{G} .

$\text{REP}_\mu^\pi(N, G)$ ist Q -kosteneindeutig.

Beweis (Lemma 103)

Seien w, w' Abläufe von $\text{REP}_\mu^\pi(N, G)$. Resultiere w in m und w' in m' . Gelte $m|_Q = m'|_Q$. Sei $x = \mu(\llbracket w|_{T'} \rrbracket_N)$ und $x' = \mu(\llbracket w'|_{T'} \rrbracket_N)$. Zu zeigen $x = x'$. Da $m|_Q = m'|_Q$ gilt, überführen $w|_{T'}$ und $w'|_{T'}$ das Netz \underline{G} in dieselbe Markierung $\wr y \wr$. Da wir in $\text{REP}_\mu^\pi(N, G)$ genau dann eine Transition $t \in T'$ mit einer Kante e synchronisieren, falls sie gleiche Kosten haben, und \underline{G} aus einem Graphen von $\mathbb{K}[N, \pi]$ in *Kostenmodell* entstand, ist y mit

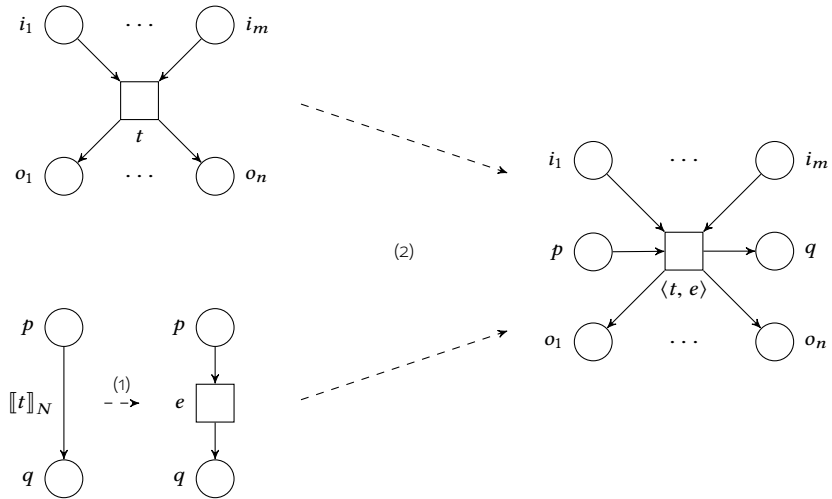


Abbildung 19 Schematisch: Die Konstruktion von $\text{REP}_\mu^\pi(N, G)$. Schritt (1) entspricht der Umwandlung von G in das Netz \underline{G} , Schritt (2) entspricht der Synchronisation von N und \underline{G} auf Basis der Kosten

einem Wert $z \in \mathbb{K}$ beschriftet und es gilt: $x = x' = z$.

Aus Lemma 35, S. 36 folgt:

Lemma 104

N und $\text{REP}_\mu^\pi(N, G)$ sind bisimilar.

Beweis (Lemma 104)

Der Graph G ist total, das heißt, für jeden Kostenwert aus N gibt es in jedem Knoten eine ausgehende Kante mit diesem Kostenwert. Da wir genau die Transitionen aus N mit den Übergängen aus \underline{G} mit denselben Kosten synchronisieren, ist die Voraussetzung von Lemma 35, S. 36 erfüllt.

Lemma 105

$$\llbracket N \rrbracket_\mu^\pi = \left[\left[\text{REP}_\mu^\pi(N, G) \right]_\mu^\pi \right]_\mu^\pi.$$

Beweis (Lemma 105)

Folgt direkt aus Lemma 104 und Gleichung (21), S. 35.

Es gilt offenbar:

Korollar 106

$\text{REP}_\mu^\pi(N, G)$ ist endlich g.d.w. die beiden folgenden Aussagen gelten:

- (1) N ist endlich.
- (2) \underline{G} ist endlich.

Die Existenz eines endlichen Graphen G von $\mathbb{K}[N, \pi]$ in μ ist jedoch im Allgemeinen unentscheidbar (siehe Lemma 66, S. 54). Ist μ assoziativ, existiert jedoch ein kanonischer Graph $G_{\mu, \mathbb{K}[N, \pi]}$ von $\mathbb{K}[N, \pi]$ in μ . Daher führen wir für assoziative Kostenmodelle die folgende Notation ein:

Notation 107 (Kosteneindeutiges, äquivalentes Netz (assoziativ))

Sei μ assoziativ. Wir schreiben $\text{REPAssoc}_\mu^\pi(N)$ für $\text{REP}_\mu^\pi(N, G_{\mu, \mathbb{K}[N, \pi]})$.

Zudem können wir die folgende Aussage auf Basis von Korollar 70, S. 56 treffen:

Korollar 108

Sei μ assoziativ. Dann ist $\text{REPAssoc}_\mu^\pi(N)$ endlich g.d.w. die beiden folgenden Aussagen gelten:

- (1) N ist endlich.
- (2) $(\mathbb{K}[N, \pi])^\mu$ ist endlich.

Fügen wir die vorherigen Aussagen zusammen, erhalten wir:

Satz 109 (Äquivalentes, kosteneindeutiges Netz)

Sei N endlich. Sei μ assoziativ. Sei Q die Menge der Zustände von $G_{\mu, \mathbb{K}[N, \pi]}$. Sei $(\mathbb{K}[N, \pi])^\mu$ endlich. Dann hat $\text{REPAssoc}_\mu^\pi(N)$ die folgenden Eigenschaften:

- (1) $\text{REPAssoc}_\mu^\pi(N)$ ist endlich.
- (2) $\text{REPAssoc}_\mu^\pi(N)$ ist Q -kosteneindeutig.
- (3) $\llbracket \text{REPAssoc}_\mu^\pi(N) \rrbracket_\mu^\pi = \llbracket N \rrbracket_\mu^\pi$.
- (4) $\text{REPAssoc}_\mu^\pi(N)$ und N sind bisimilar.

Beweis (Satz 109)

Folgt aus Korollar 108, Lemma 103, S. 75, Lemma 104, S. 76 und Lemma 105, S. 76.

3.5.1.3 Alternative Konstruktion für Filter-Perspektiven und kommutative Verbands-Kostenmodelle

Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei N ein gewichtetes Netz mit Transitionen T . Sei $T \subseteq T'$ und $\pi = \text{FILTER}[T']$. Sei $\mathbb{K}[N, \pi]$ kommutativ in μ . Wir stellen eine Konstruktion eines zu N kostenäquivalenten, kosteneindeutigen Netzes vor.

Sei $\text{REP}\text{KOMM}_\mu^\pi(N)$ das Netz, das aus N wie folgt entsteht: Für jedes $x \in \mathbb{K}[N, \pi]$ fügen wir x als Platz ein, sodass jede Transition $t \in T'$ mit Kosten x genau eine Marke auf x produziert.

Da wir Plätze hinzufügen, kann $\text{REP}\text{KOMM}_\mu^\pi(N)$ nur endlich sein, wenn N auch endlich ist. Ist N endlich, ist auch T' und damit auch $\mathbb{K}[N, \pi]$ endlich. Also ist dann $\text{REP}\text{KOMM}_\mu^\pi(N)$ auch endlich.

Korollar 110

N ist endlich g.d.w. $\text{REP}\text{KOMM}_\mu^\pi(N)$ endlich ist.

Lemma 111

$\text{REP}\text{KOMM}_\mu^\pi(N)$ ist $\mathbb{K}[N, \pi]$ -kosteneindeutig.

Beweis (Lemma 111)

Seien w, w' Abläufe von $\text{REP}\text{KOMM}_\mu^\pi(N)$. Resultiere w in m und w' in m' . Gelte $m|_{\mathbb{K}[N, \pi]} = m'|_{\mathbb{K}[N, \pi]}$. Sei $x = \mu(\llbracket w|_{T'} \rrbracket_N)$ und $x' = \mu(\llbracket w'|_{T'} \rrbracket_N)$. Zu zeigen $x = x'$. Da $m|_{\mathbb{K}[N, \pi]} = m'|_{\mathbb{K}[N, \pi]}$ gilt, sind $w|_{T'}$ und $w'|_{T'}$ Permutationen voneinander. Da μ kommutativ ist, gilt also $x = x'$.

Lemma 112

N und $\text{REP}\text{KOMM}_\mu^\pi(N)$ sind bisimilar.

Beweis (Lemma 112)

Dies folgt direkt daraus, dass wir nur weitere Nachplätze eingefügt haben. Die Plätze aus $\mathbb{K}[N, \pi]$ haben also keinerlei Einfluss auf die Schaltregel.

Lemma 113

$$\llbracket N \rrbracket_\mu^\pi = \left[\left[\text{REP}\text{KOMM}_\mu^\pi(N) \right]_\mu^\pi \right]_\mu^\pi.$$

Beweis (Lemma 113, S. 78)

Folgt direkt aus Lemma 112, S. 78 und Gleichung (21), S. 35.

Fügen wir die vorherigen Aussagen zusammen, erhalten wir:

Satz 114 (Äquivalentes, kosteneindeutiges Netz)

Sei N endlich. Sei μ kommutativ. Sei $(\mathbb{K}[N, \pi])^\mu$ endlich. Dann hat das Netz $\text{REPKOMM}_\mu^\pi(N)$ die folgenden Eigenschaften:

- (1) $\text{REPKOMM}_\mu^\pi(N)$ ist endlich.
- (2) $\text{REPKOMM}_\mu^\pi(N)$ ist $\mathbb{K}[N, \pi]$ -kosteneindeutig.
- (3) $\llbracket \text{REPKOMM}_\mu^\pi(N) \rrbracket_\mu^\pi = \llbracket N \rrbracket_\mu^\pi$.
- (4) $\text{REPKOMM}_\mu^\pi(N)$ und N sind bisimilar.

Beweis (Satz 114)

Folgt aus Korollar 110, S. 78, Lemma 111, S. 78, Lemma 112, S. 78 und Lemma 113, S. 78.

3.5.1.4 Alternative Konstruktion für Filter-Perspektiven und die Max-Plus-Algebra

Wir betrachten das von der *Max-Plus-Algebra* $\text{MP} = \langle \mathbb{N}_0 \cup \{-\infty\}, \max, + \rangle$ induzierte Kostenmodell μ , sowie eine Filter-Perspektive $\pi = \text{FILTER}[T']$. Wir zeigen, dass zu jedem endlichen Netz N ein kostenäquivalentes, kosteneindeutiges Netz N' existiert. Dabei betrachten wir nur solche Netze, in denen an keiner Transition Kosten von ω stehen.

Um die Kosteneindeutigkeit zu garantieren, kodieren wir die Kosten eines Ablaufs in seinem resultierenden Zustand. Dabei unterscheiden wir generell zwei Fälle: Im ersten Fall enthält der Ablauf keine Transition mit Kosten $-\infty$. Im zweiten Fall enthält der Ablauf wenigstens eine $-\infty$ -Transition. Im ersten Fall entsprechen die Kosten einer natürlichen Zahl, im zweiten $-\infty$. Wir berücksichtigen bei der Kodierung beide Fälle, indem wir uns eine natürliche Zahl sowie die Information merken, ob eine $-\infty$ -Transition gefeuert hat. Wir stellen dabei sicher, dass im ersten Fall die natürliche Zahl stets den tatsächlichen Kosten des Ablaufs entspricht.

Um zwischen beiden Fällen unterscheiden zu können, duplizieren wir die Transitionen des Netzes: Für jede Transition t gibt es je die Transition t , die nur eintreten kann, *bevor* eine Transition mit Kosten $-\infty$ eingetreten

ist, und zweitens eine Transition \tilde{t} , die nur eintreten kann, *nachdem* eine Transition mit Kosten $-\infty$ eingetreten ist. Die beiden Zustände realisieren wir durch die Plätze \hat{p}_{ctrl} (vor Schalten einer $-\infty$ -Transition) und $\hat{p}_{-\infty}$ (nach dem Schalten wenigstens einer $-\infty$ -Transition). Da neu entstehende Kosten zu den bestehenden Kosten addiert werden, können wir die natürliche Zahl bequem als Marken auf einem ausgezeichneten Platz \hat{p} speichern.

Wir beschreiben die schematisch in Abbildung 20 dargestellte Konstruktion: Sei dazu N ein Netz mit Transitionen $T, T' \subseteq T$ und $\pi = \text{FILTER}[T']$. Wir führen in das bestehende Netz N drei neue Plätze ein, die wir mit $\hat{p}, \hat{p}_{-\infty}$ und \hat{p}_{ctrl} notieren. Initial sei \hat{p}_{ctrl} einfach markiert und $\hat{p}, \hat{p}_{-\infty}$ leer. Jede Transition $t \in T'$ mit Kosten $k \in \mathbb{N}_0$ hat eine Schlinge zu \hat{p}_{ctrl} und produziert k Marken auf den Platz \hat{p} . Jede Transition $t \in T'$ mit $k = -\infty$ konsumiert eine Marke von \hat{p}_{ctrl} und produziert eine Marke auf $\hat{p}_{-\infty}$. Anschließend fügen wir für jede Transition $t \in T'$ eine weitere Transition ein, die wir als \tilde{t} bezeichnen. Bezüglich der Plätze von N verhält sich \tilde{t} genauso wie t , das heißt, für jeden Platz p von N gilt: \tilde{t} konsumiert (produziert) von (auf) p genauso viele Marken wie t . Jede Transition \tilde{t} hat außerdem eine Schlinge zu $\hat{p}_{-\infty}$.

Wir bezeichnen das auf diese Weise konstruierte Petrinetz als $\text{REP}_{\text{MP}}^{T'}(N)$. Da die konkrete Wahl der Plätze $\hat{p}, \hat{p}_{\text{ctrl}}$ und $\hat{p}_{-\infty}$ unerheblich ist, erscheinen sie auch nicht als Parameter in der Schreibweise.

Lemma 115

$\text{REP}_{\text{MP}}^{T'}(N)$ ist $\{\hat{p}, \hat{p}_{-\infty}\}$ -kosteneindeutig bezüglich π und μ .

Beweis (Lemma 115)

Sei $w = t_1 \dots t_n$ ein in einer Markierung m resultierender Ablauf.

- (1) Gelte für alle $1 \leq i \leq n$: Die Kosten von t_i sind ungleich $-\infty$. Dann gilt $m(\hat{p}_{\text{ctrl}}) = 1$ und $m(\hat{p}_{-\infty}) = 0$. Außerdem liegen auf \hat{p} Marken in Höhe der Kosten von w : Jede Transition t_i legt genau die Kosten von t_i auf den Platz \hat{p} und keine Transition konsumiert Marken von \hat{p} .
- (2) Gelte für wenigstens ein $1 \leq i \leq n$: Die Kosten von t_i sind $-\infty$. Sei m_i für alle $1 \leq i \leq n$ die aus $t_1 \dots t_i$ resultierende Markierung. Sei k der kleinste Index mit $1 \leq k \leq n$, so dass für alle $1 \leq j \leq k$ gilt: t_j hat Kosten ungleich $-\infty$. Dann gilt wie in Fall 1 erklärt für m_k : $m_k(\hat{p}_{\text{ctrl}}) = 1$ und $m_k(\hat{p}_{-\infty}) = 0$. Da die Transition t_{k+1} Kosten von $-\infty$ aufweist und $m_k(\hat{p}_{\text{ctrl}}) = 1$ gilt, gilt $m_{k+1}(\hat{p}_{-\infty}) = 1$ und $m_{k+1}(\hat{p}_{\text{ctrl}}) = 0$. Ab jetzt bleibt der Platz $\hat{p}_{-\infty}$ markiert. Die Kosten von w betragen offenbar $-\infty$.

Also ist $\text{REP}_{\text{MP}}^{T'}(N)$ $\{\hat{p}, \hat{p}_{-\infty}\}$ -kosteneindeutig.

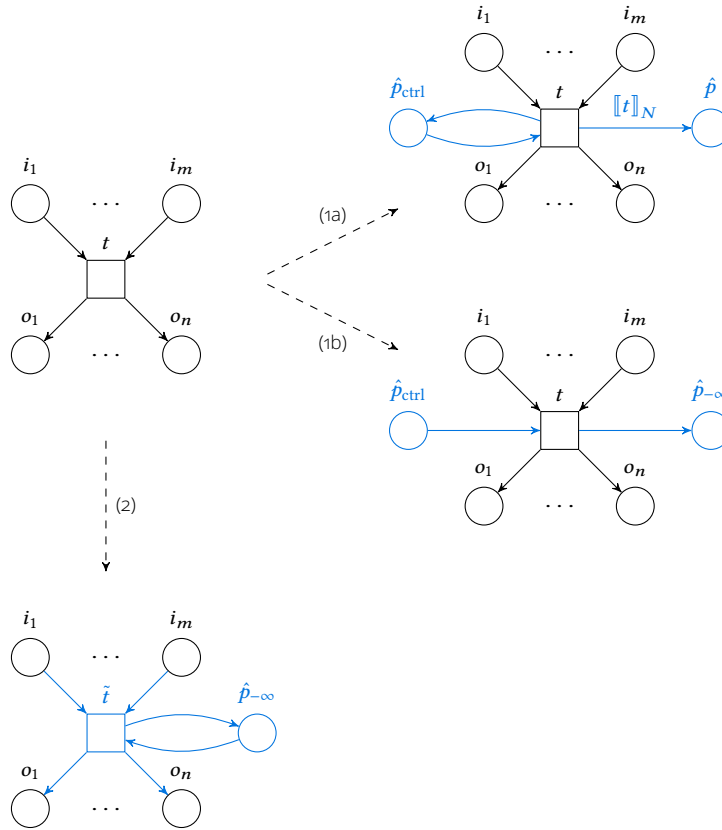


Abbildung 20 Schematisch: Die Konstruktion von $\text{REP}_{\text{MP}}^{T'}(N)$. Alle neu hinzugefügten Elemente sind blau. Schritt (1a) tritt nur ein, falls $\llbracket t \rrbracket_N \in \mathbb{N}_0$. Andernfalls tritt Schritt (1b) ein. Die Gewichte der von i_1, \dots, i_m ausgehenden sowie der in o_1, \dots, o_n eingehenden Kanten bleiben bei der Konstruktion stets erhalten.

Außerdem können wir zeigen, dass N und $\text{REP}_{\text{MP}}^{T'}(N)$ bisimilar und damit kostenäquivalent sind:

Lemma 116

N und N' sind bisimilar.

Beweis (Lemma 116)

Wir können den Beweis sehr ähnlich wie für Lemma 104, S. 76 führen. Die einzige Schwierigkeit besteht in der jeweils neuen Transition \tilde{t} für jede Transition t . Sei $N' = \text{REP}_{\text{MP}}^{T'}(N)$. Sei R die folgende Relation auf Markierungen von N und N' : Sei m eine Markierung von N und m' eine Markierung von N' . Dann gelte genau dann $\langle m, m' \rangle \in R$, wenn $m = m'|_P$ gilt. Dabei sei P die Menge der Plätze von N . Wir zeigen, dass

R eine Simulationsrelation ist. Sei $\langle m_1, m'_1 \rangle \in R$ und $m_1 \xrightarrow{t} m_2$. Dann gilt $m_1 = m'_1|_P$.

- (1) Gelte $m'_1(\hat{p}_{\text{ctrl}}) = 1$. Dann gilt $m'_1(\hat{p}_{-\infty}) = 0$. Dann ist t auch in m'_1 aktiviert und es gilt $m'_1 \xrightarrow{t} m'_2$. Da t in N und N' die Plätze aus P genau gleich manipuliert, gilt $m_2 = m'_2|_P$ und damit auch $\langle m_2, m'_2 \rangle \in R$.
- (2) Gelte $m'_1(\hat{p}_{\text{ctrl}}) \neq 1$. Dann gilt $m'_1(\hat{p}_{\text{ctrl}}) = 0$ und $m'_1(\hat{p}_{-\infty}) = 1$. Dann ist die Transition \tilde{t} in m'_1 aktiviert und es gilt $m'_1 \xrightarrow{\tilde{t}} m'_2$. Da \tilde{t} in N' die Plätze in P auf die gleiche Weise wie t in N manipuliert, gilt $m_2 = m'_2|_P$ und damit auch $\langle m_2, m'_2 \rangle \in R$.

Da die Beschriftung oder die gesendeten/empfangenen Nachrichten einer Transition bei der Konstruktion nicht verändert werden, ist R eine Simulationsrelation. Analog können wir zeigen, dass R^{-1} ebenfalls eine Simulationsrelation ist. Zuletzt können wir beobachten, dass R initial ist: Die Anfangsmarkierung von N' ist die Anfangsmarkierung von N zusammen mit einer Marke auf \hat{p}_{ctrl} . Also ist R eine initiale Bisimulationsrelation und N und N' sind bisimilar.

Lemma 117

$$\llbracket N \rrbracket_{\text{MP}}^\pi = \llbracket \text{REP}_{\text{MP}}^{T'}(N) \rrbracket_{\text{MP}}^\pi.$$

Beweis (Lemma 117)

Folgt aus Lemma 116, S. 81 und der Tatsache, dass t und \tilde{t} stets die gleichen Kosten aufweisen.

Wir betrachten die Bedingungen, unter denen der Platz \hat{p} in $\text{REP}_{\text{MP}}^{T'}(N)$ beschränkt ist:

Lemma 118

Die folgenden Aussagen sind äquivalent:

- (1) $\llbracket N \rrbracket_\mu^\pi \neq \omega$.
- (2) \hat{p} ist beschränkt in $\text{REP}_{\text{MP}}^{T'}(N)$.

Beweis (Lemma 118)

1. \Rightarrow 2.: Da $\llbracket N \rrbracket_\mu^\pi \neq \omega$, gibt es nur endlich viele verschiedene Kostenwerte die an Abläufen auftreten. Daher ist \hat{p} beschränkt.

2. \Rightarrow 1.: Da \hat{p} beschränkt ist, können nur endlich viele verschiedene Kos-

tenwerte auftreten. Also gilt $\llbracket N \rrbracket_\mu^\pi \neq \omega$.

Wir betrachten die Umstände, unter denen $\text{REP}_{\text{MP}}^{T'}(N)$ beschränkt ist:

Lemma 119

Die folgenden Aussagen sind äquivalent:

- (1) N ist beschränkt und $\llbracket N \rrbracket_\mu^\pi \neq \omega$.
- (2) $\text{REP}_{\text{MP}}^{T'}(N)$ ist beschränkt.

Beweis (Lemma 119)

Die Beschränktheit der Plätze \hat{p}_{ctrl} und $\hat{p}_{-\infty}$ ist klar an der Struktur erkennbar. Die Beschränktheit des Platzes \hat{p} korreliert laut Lemma 118, S. 82 mit $\llbracket N \rrbracket_\mu^\pi \neq \omega$. Die Beschränktheit der Plätze aus N ändert sich nicht durch die Konstruktion.

Zuletzt stellen wir fest, dass die Konstruktion genau dann ein endliches Resultat liefert, falls N endlich ist.

Korollar 120

N ist endlich g.d.w. $\text{REP}_{\text{MP}}^{T'}(N)$ endlich ist.

Fügen wir die vorherigen Aussagen zusammen, erhalten wir:

Satz 121 (Äquivalentes, kosteneindeutiges Netz)

Sei N endlich. Dann hat $\text{REP}_{\text{MP}}^{T'}(N)$ die folgenden Eigenschaften:

- (1) $\text{REP}_{\text{MP}}^{T'}(N)$ ist endlich.
- (2) $\text{REP}_{\text{MP}}^{T'}(N)$ ist $\{\hat{p}, \hat{p}_{-\infty}\}$ -kosteneindeutig.
- (3) $\llbracket \text{REP}_{\text{MP}}^{T'}(N) \rrbracket_\mu^\pi = \llbracket N \rrbracket_\mu^\pi$.
- (4) $\text{REP}_{\text{MP}}^{T'}(N)$ und N sind bisimilar.

Beweis (Satz 121)

Folgt aus Korollar 120, Lemma 115, S. 80, Lemma 116, S. 81 und Lemma 117, S. 82.

3.5.2 Kosteneindeutige äquivalente Netze bis zu einer Schranke

Wir betrachten im Folgenden ein gewichtetes Netz N über \mathbb{K} mit Transitionen T . Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Dann induziert jedes Element $k \in \mathbb{K}$ eine Teilmenge $\mathbb{K}_{\leq_\mu k} := \{a \mid a \in \mathbb{K}, a \leq_\mu k\}$. Daraus ergibt sich die folgende Definition:

Definition 122 (k -Kosteneindeutigkeit)

Sei $k \in \mathbb{K}$. Sei N $\mathbb{K}_{\leq_\mu k}$ -kosteneindeutig bzgl. π und μ . Dann ist N k -kosteneindeutig bzgl. π und μ .

Wir zeigen zunächst die Existenz eines äquivalenten, k -kosteneindeutigen Netzes auf Basis der Existenz eines äquivalenten kosteneindeutigen Netzes. Anschließend zeigen wir eine Konstruktion für eine Teilklasse von Kostenmodellen vor.

3.5.2.1 Existenz k -kosteneindeutiger, äquivalenter Netze

Jedes kosteneindeutige Netz ist offenbar auch k -kosteneindeutig. Also können wir aus Korollar 101, S. 74 direkt folgern:

Korollar 123

Zu jedem Netz N und jedem $k \in \mathbb{K}$ existiert ein k -kosteneindeutiges Netz N' mit $\llbracket N \rrbracket_\mu^\pi = \llbracket N' \rrbracket_\mu^\pi$.

3.5.2.2 Konstruktion für Verbands-Kostenmodelle

Wir betrachten ein Verbands-Kostenmodell μ über \mathbb{K} , sowie eine Filter-Perspektive $\pi = \text{FILTER}[T']$. Sei k eine Schwelle von $\mathbb{K}[N, \pi]$ in μ .

Sei \underline{G} ein Netz, das aus einem k -Graphen G von $\mathbb{K}[N, \pi]$ in μ mit Wurzelzustand r entsteht. Dabei ist $\lceil r \rceil$ die Anfangsmarkierung von \underline{G} .

Sei $\text{REP}_\mu^\pi[k](N, G)$ das synchrone Produkt aus N und \underline{G} , wobei wir eine Transition $t \in T'$ genau dann mit einer Transition e von \underline{G} synchronisieren, wenn e die Beschriftung $\llbracket t \rrbracket_N$ in G aufweist (siehe Abschnitt 2.7.7, S. 35).

Wir zeigen die folgenden Aussagen analog zu denen aus Abschnitt 3.5.1.2, S. 75:

Lemma 124

Sei Q die Menge der Zustände von $A_{N, \mu} \cdot \text{REP}_\mu^\pi[k](N, G)$ ist Q , k -kosteneindeutig.

Beweis (Lemma 124, S. 84)

Können wir völlig analog zu Lemma 103, S. 75 führen.

Lemma 125

N und $\text{REP}_\mu^\pi[k](N, G)$ sind bisimilar.

Beweis (Lemma 125)

Wir können den Beweis hier völlig analog zu dem von Lemma 104, S. 76 führen.

Lemma 126

$$\llbracket N \rrbracket_\mu^\pi = \left\llbracket \text{REP}_\mu^\pi[k](N, G) \right\rrbracket_\mu^\pi.$$

Beweis (Lemma 126)

Folgt direkt aus Lemma 125 und Gleichung (21), S. 35.

Es gilt offenbar:

Korollar 127

$\text{REP}_\mu^\pi[k](N, G)$ ist endlich g.d.w. die beiden folgenden Aussagen gelten:

- (1) N ist endlich.
- (2) \underline{G} ist endlich.

Die Existenz eines endlichen k -Graphen G von $\mathbb{K}[N, \pi]$ in μ ist jedoch laut Lemma 72, S. 57 unentscheidbar. Für assoziative Kostenmodelle existiert immer der kanonische k -Graph, weswegen wir die folgende Notation einführen (analog zu Notation 107, S. 77):

Notation 128 (Kanonisches kosteneindeutiges Netz)

Sei μ assoziativ. Wir schreiben $\text{REPAssoc}_\mu^\pi[k](N)$ für das Netz

$$\text{REP}_\mu^\pi[k](N, G_{\mu, \mathbb{K}[N, \pi], k}) .$$

Zudem können wir die folgende Aussage auf Basis von Korollar 76, S. 59 treffen:

Korollar 129

Sei μ assoziativ. Dann ist $\text{REPAssoc}_\mu^\pi[k](N)$ endlich g.d.w. die beiden folgenden Aussagen gelten:

- (1) N ist endlich.
- (2) $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ ist endlich.

Satz 130 (Äquivalentes, kosteneindeutiges Netz)

Sei N endlich. Sei μ assoziativ. Sei Q die Menge der Zustände von $G_{\mu, \mathbb{K}[N, \pi]}$. Sei $(\mathbb{K}[N, \pi])^\mu$ endlich. Dann hat $\text{REPAssoc}_\mu^\pi[k](N)$ die folgenden Eigenschaften:

- (1) $\text{REPAssoc}_\mu^\pi[k](N)$ ist endlich.
- (2) $\text{REPAssoc}_\mu^\pi[k](N)$ ist Q, k -kosteneindeutig.
- (3) $\left[\left[\text{REPAssoc}_\mu^\pi[k](N) \right]_\mu^\pi \right]_\mu^\pi = \llbracket N \rrbracket_\mu^\pi$.
- (4) $\text{REPAssoc}_\mu^\pi[k](N)$ und N sind bisimilar.

Beweis (Satz 130)

Folgt aus Korollar 129, Lemma 124, S. 84, Lemma 125, S. 85 und Lemma 126, S. 85.

3.6 Anforderungen

Anforderungen

In diesem Abschnitt formalisieren wir die in Abschnitt 1.2.2, S. 6 abstrakt eingeführten *Anforderungen*. Einige dieser Anforderungen beziehen sich auf beliebige Netze, andere ausdrücklich auf gewichtete Netze über einer Menge von Kostenwerten. Formal können wir eine Anforderung als eine Menge von Netzen mit einer bestimmten Eigenschaft auffassen: Ein Netz erfüllt dann die Anforderung, falls es in der Menge enthalten ist. Da jedes gewichtete Netz ein Netz ist, gelingt dieser Ansatz auch, falls die Anforderung eigentlich nur für gewichtete Netze über einer bestimmten Menge von Kostenwerten definiert ist: Jedes andere Netz erfüllt dann die Anforderung nicht.

Notation 131 (Durchschnitt von Anforderungen)

Häufig betrachten wir den Durchschnitt zweier Anforderungen ρ_1 und ρ_2 . Wo es die Lesbarkeit steigert, schreiben wir $\rho_1 \rho_2$ anstatt $\rho_1 \cap \rho_2$.

3.6.1 Strukturelle Anforderungen

Wir beginnen mit der einzigen in dieser Arbeit behandelten strukturellen Anforderung: der *Geschlossenheit*. Diese bezieht sich auf die Schnittstelle (siehe Definition 27, S. 32) eines gewichteten Netzes. Jedes Netz ohne ausgezeichnete Schnittstelle können wir dabei einfach als offenes Netz mit der leeren Schnittstelle ansehen.

Definition 132 (Geschlossen)

Ein Netz heißt *geschlossen*, falls seine Schnittstelle leer ist.

3.6.2 Funktionale Verhaltensanforderungen

In diesem Abschnitt betrachten wir die Anforderungen *Beschränktheit* und *schwache Terminierung*.

3.6.2.1 Beschränktheit

Die nächste betrachtete Anforderung ist die *Beschränktheit*, die sich auf die Menge der erreichbaren Markierungen bezieht.

Definition 133 (Beschränktheit)

Ein Netz N ist k -beschränkt, falls für alle erreichbaren Markierungen m und jeden Platz p von N gilt: $m(p) \leq k$. Ein Netz ist *beschränkt*, falls es k -beschränkt für wenigstens ein k ist.

Beispiel 134

Das Netz Net_{13} aus Abbildung 13, S. 44 ist 1-beschränkt, das Netz Net_{18} aus Abbildung 18, S. 70 ist unbeschränkt.

Aus Definition 133 folgt direkt:

Korollar 135

Ein Netz N ist beschränkt g.d.w. $\text{REACH}(N)$ eine endliche Menge ist. Ein Netz N mit n Plätzen ist k -beschränkt g.d.w. $\text{REACH}(N)$ höchstens $(k+1)^n$ Elemente enthält.

3.6.2.2 Schwache Terminierung

Die Anforderung der *schwachen Terminierung* betrachtet die Rahmenmarkierungen und Zielmarkierungen eines Netzes. Konkret fordert schwache Terminierung, dass ausschließlich Rahmenmarkierungen erreichbar sind und von jeder Markierung eine Zielmarkierung erreichbar ist. In dieser Arbeit betrachten wir der Einfachheit halber die Anforderung der schwachen Terminierung nur in Zusammenhang mit der Anforderung der Geschlossenheit.

schwach terminierend

Definition 136 (Schwache Terminierung)

Sei N ein geschlossenes Petrinetz mit Transitionen T und ausgezeichneten Mengen M^\square und M^\diamond von Rahmenmarkierungen und Zielmarkierungen. Dann heißt N *schwach terminierend* g.d.w.

- (1) $\text{REACH}(N) \subseteq M^\square$ und
- (2) für alle Abläufe $v \in \text{RUNS}(N)$ existiert ein $w \in T^*$ mit $vw \in \text{RUNS}(N, M^\diamond)$.

Beispiel 137

Wir betrachten das Netz Net_{13} aus Abbildung 13, S. 44 und einige Varianten für Rahmenmarkierungen und Zielmarkierungen:

- Sei $\{m \mid m(p) \leq 1 \text{ für alle } p\}$ die Menge der Rahmenmarkierungen und die Anfangsmarkierung die einzige Zielmarkierung. Dann ist Net_{13} schwach terminierend.
- Sei $\{m \mid \sum_{\text{Platz } p} m(p) \leq 2\}$ und die Menge der Zielmarkierungen beliebig gewählt. Dann ist Net_{13} *nicht* schwach terminierend, da nach Feuern der Transition t_1 die Plätze P_2 , P_3 und B markiert sind.
- Ist die Menge der Zielmarkierungen leer, ist Net_{13} *nicht* schwach terminierend, da von keiner erreichbaren Markierung eine Zielmarkierung erreichbar ist.
- Seien die Menge der Rahmenmarkierungen eine Obermenge und die Menge der Zielmarkierungen eine nicht-leere Teilmenge der erreichbaren Markierungen von Net_{13} . Dann ist Net_{13} schwach terminierend, da in Net_{13} für zwei erreichbare Markierungen m und m' stets gilt: m ist aus m' erreichbar. Daher sind nur Rahmenmarkierungen erreichbar und von jeder erreichbaren Markierung ist eine Zielmarkierung erreichbar.

Das Netz Net_{14} aus Abbildung 14, S. 44 ist dagegen nur dann schwach terminierend, falls $m = \{P_2, B\}$ eine Zielmarkierung ist: Die Markierung m ist erreichbar und es ist keine weitere Markierung aus m erreichbar.

Wir führen die folgende Notation ein:

Notation 138 (Menge der schwach terminierenden)

Wir schreiben \mathbb{S} für die Menge der Netze, die geschlossen und schwach terminierend sind.

\mathbb{S}

3.6.3 Nicht-funktionale Verhaltensanforderungen

Wir betrachten in dieser Arbeit die folgenden nicht-funktionalen Verhaltensanforderungen für gewichtete Netze:

- *k-Kostenbeschränktheit* fordert, dass die Kosten des Netzes unter einer Schranke k liegen.
- *Kostenbeschränktheit* fordert, dass eine Schranke existiert, unter der die Kosten des Netzes liegen.

k-Kostenbeschränktheit

Kostenbeschränktheit

Dazu halten wir eine Perspektive π sowie ein Kostenmodell μ über \mathbb{K} fest. Beschränktheit einer Menge X bedeutet allgemein die Existenz eines Wertes y , sodass alle Werte $x \in X$ *kleiner gleich* y sind. Dieses *kleiner gleich* entnehmen wir einer *Präferenzrelation* auf den Kostenwerten. Eine Präferenzrelation ist laut Jehle und Reny [JR11] eine binäre, totale und transitive Relation (totale Quasiordnung) \leq auf \mathbb{K} . Häufig werden weitere Eigenschaften angenommen, mit denen Effekte erreicht werden, die der Antisymmetrie ähneln. Daher ähneln die in [JR11] definierten Präferenzrelationen totalen Ordnungen. Das Kreuzprodukt zweier totaler Ordnungen ist im Allgemeinen nicht total. Um trotzdem das Kreuzprodukt zweier Präferenzrelationen als Präferenzrelation auffassen zu können, betrachten wir in dieser Arbeit vollständige Verbände, die erstens laut Lemma 3, S. 20 abgeschlossen unter dem Kreuzprodukt sind, und zweitens der Voraussetzung der Totalordnung am nächsten kommen: Zwei Kostenwerte sind nicht unbedingt vergleichbar, es existiert jedoch stets eine kleinste obere Schranke dieser Werte. Die Vollständigkeit induziert zudem die Existenz einer *größten Schranke* für alle Kostenwerte: Das Supremum \top von \mathbb{K} . Diese Schranke erlaubt eine einfache Definition des Begriffes der Kostenbeschränktheit. Sei $\langle \mu, \leq \rangle$ ein vollständiger Verband¹ mit größtem Element \top . Ein gewichtetes Netz S über \mathbb{K} ist *k-kostenbeschränkt* für ein $k \neq \top \in \mathbb{K}$, falls für die Kosten von S durch k beschränkt werden. *Kostenbeschränktheit* fordert dagegen die Existenz wenigstens eines solchen k .

¹Dies impliziert nicht, dass μ auch über $\langle \mathbb{K}, \leq \rangle$ definiert ist.

Definition 139 (k-Kostenbeschränktheit, Kostenbeschränktheit)

Sei S ein gewichtetes Netz über \mathbb{K} . Sei π eine Perspektive von S . Sei μ ein Kostenmodell über \mathbb{K} . Sei $\langle \mathbb{K}, \leq \rangle$ ein vollständiger Verband mit größtem Element \top .

- (1) Sei $k \neq \top \in \mathbb{K}$ mit $\llbracket S \rrbracket_{\mu}^{\pi} \leq k$. Dann ist S k -kostenbeschränkt bezüglich π , μ und \leq .
- (2) Sei $\llbracket S \rrbracket_{\mu}^{\pi} \neq \top$. Dann ist S kostenbeschränkt bezüglich π , μ und \leq .

Notation 140 (Menge der kostenbeschränkten Netze)

Wir schreiben $\mathbb{B}_{\mu, \leq}^{\pi}[k]$ für die Menge der k -kostenbeschränkten Netze. Analog dazu schreiben wir $\mathbb{B}_{\mu, \leq}^{\pi}$ für die kostenbeschränkten Netze. Ist μ über $\langle \mathbb{K}, \leq \rangle$ definiert, entfällt der Parameter \leq und wir schreiben auch $\mathbb{B}_{\mu}^{\pi}[k]$ für $\mathbb{B}_{\mu, \leq}^{\pi}[k]$ und \mathbb{B}_{μ}^{π} für $\mathbb{B}_{\mu, \leq}^{\pi}$.

$$\mathbb{B}_{\mu, \leq}^{\pi}[k]$$

$$\mathbb{B}_{\mu, \leq}^{\pi}$$

Beispiel 141

Wir betrachten die Netze Net_{13} aus Abbildung 13, S. 44, Net_{15} aus Abbildung 15, S. 67, Net_{17} aus Abbildung 17, S. 68 und Net_{16} aus Abbildung 16, S. 68 unter dem Kostenmodell der Max-Plus-Algebra und unterschiedlichen Perspektiven.

- Net_{13} ist kostenbeschränkt unter der Perspektive $\text{FILTER}[\{t_1, t_3\}]$ jedoch *nicht* kostenbeschränkt unter der Perspektive $\text{FILTER}[\{t_1, t_2, t_3\}]$.
- Net_{15} ist kostenbeschränkt unter der Perspektive $\text{FILTER}[\{t_1, t_2, t_3\}]$, jedoch *nicht* kostenbeschränkt unter der Perspektive, die jedes Vorkommen von t_1 durch ein Vorkommen von t_2 ersetzt.
- Für jedes $k \in \mathbb{N}_0$ ist Net_{16} k -kostenbeschränkt unter der Perspektive $\text{FILTER}[t_1, t_2, t_3]$.
- Net_{17} ist kostenbeschränkt unter der Perspektive $\text{FILTER}[\{t_1, t_3\}]$, jedoch *nicht* kostenbeschränkt unter der Perspektive $\text{FILTER}[\{t_1, t_2, t_3\}]$.
- Sei $\Sigma = \{b, n, r\}$ und U der Anwendungsfall mit $\bullet U = \Sigma^* \{b\}$ und $U^{\bullet} = \{n, r\}$. Sei für $N \in \{\text{Net}_{13}, \text{Net}_{15}, \text{Net}_{17}, \text{Net}_{16}\}$ jeweils N' die Variante, in der die Transitionen die folgende Beschriftung aufweisen: $t_1 \mapsto b$, $t_2 \mapsto n$ und $t_3 \mapsto r$ und $t \mapsto \varepsilon$ für die übrigen Transitionen. Dann überdecken alle N den Anwendungsfall U . Außerdem sind alle N kostenbeschränkt unter der Perspektive $\text{FILTER}[\{t_1, t_2, t_3\}] \triangleright \text{AF}[N, U]$. Außerdem ist N' 0-kostenbeschränkt, falls $N \in \{\text{Net}_{15}, \text{Net}_{16}\}$ für $k = 0$ und 1-kostenbeschränkt, falls $N \in \{\text{Net}_{13}, \text{Net}_{17}, \text{Net}_{16}\}$ für $k = 1, 2, \dots\}$.

3.7 Präferenzen

Eine *Präferenz* \succsim ist im Rahmen dieser Arbeit eine Quasiordnung auf Netzen. Dabei lässt sich $N \succsim N'$ wie folgt lesen: N ist *wenigstens so gut wie* N' . Gilt zusätzlich $N' \not\succsim N$, wird N über N' *präferiert*. Gelten $N \succsim N'$ und $N' \succsim N$, sind wir bezüglich N und N' *indifferent*. Gilt weder $N \succsim N'$ noch $N' \succsim N$, sind N und N' *unvergleichbar*. In diesem Abschnitt führen wir erst die in Abschnitt 1.2.2, S. 6 abstrakt eingeführte Präferenz *kostengünstiger* auf gewichteten Netzen über einer Menge \mathbb{K} ein. Anschließend definieren wir die kanonische *Indifferenz* auf Netzen.

Präferenz

Wie schon im vorherigen Abschnitt Abschnitt 3.6.3, S. 89 stützen wir unsere Definitionen auf eine gegebene vollständige Verbandsordnung \leq auf \mathbb{K} . Diese Relation übertragen wir auf die Menge der gewichteten Netze über \mathbb{K} : Sei $\langle \mathbb{K}, \leq \rangle$ ein vollständiger Verband². Die Präferenz *kostengünstiger* überträgt die Ordnung \leq wie folgt auf gewichtete Netze über \mathbb{K} : Ein Netz S_1 wird über ein Netz S_2 präferiert, falls die Kosten von S_1 unter den Kosten von S_2 liegen.

Definition 142 (Kostengünstiger)

Seien S_1 und S_2 gewichtete Netze über \mathbb{K} . Dann ist S_1 *wenigstens so kostengünstig wie* S_2 bezüglich π , μ und \leq , notiert als $S_1 \succsim_{\mu, \leq}^{\pi} S_2$, falls $\llbracket S_1 \rrbracket_{\mu}^{\pi} \leq \llbracket S_2 \rrbracket_{\mu}^{\pi}$. Gelten $S_1 \succsim_{\mu, \leq}^{\pi} S_2$ und $S_2 \not\succsim_{\mu, \leq}^{\pi} S_1$, nennen wir S_1 *kostengünstiger* als S_2 .

Wir betrachten in dieser Arbeit häufig Verbands-Kostenmodelle und führen daher die folgende Notation ein:

Notation 143 (Kostengünstiger und Verbands-Kostenmodelle)

Ist μ über $\langle \mathbb{K}, \leq \rangle$ definiert, schreiben wir auch $S_1 \succsim_{\mu}^{\pi} S_2$ anstatt $S_1 \succsim_{\mu, \leq}^{\pi} S_2$.

Beispiel 144

Wir betrachten die Netze Net_{13} aus Abbildung 13, S. 44, Net_{15} aus Abbildung 15, S. 67, Net_{17} aus Abbildung 17, S. 68 und Net_{16} aus Abbildung 16, S. 68 unter dem Kostenmodell der Max-Plus-Algebra und unterschiedlichen Perspektiven.

- Betrachten wir die Perspektive $\text{FILTER}[\{t_1, t_2, t_3\}]$, beobachten wir die folgenden Beziehungen: $\text{Net}_{15}, \text{Net}_{16}$ für $k = 0 \succsim \text{Net}_{16}$ für $k = 1 \succsim \text{Net}_{16}$ für $k = 2 \succsim \dots \succsim \text{Net}_{13}, \text{Net}_{17}$
- Wir führen Beispiel 141, S. 90 fort und betrachten dazu erneut den dort definierten Anwendungsfall U sowie die dort definierten

²Das heißt nicht, dass μ über $\langle \mathbb{K}, \leq \rangle$ definiert ist

Varianten N' der Netze $N \in \{\text{Net}_{13}, \text{Net}_{15}, \text{Net}_{17}, \text{Net}_{16}\}$. Für die Perspektive $\text{FILTER}[t_1, t_2, t_3] \triangleright \text{AF}[N, U]$ sieht die Präferenz wie folgt aus: $\text{Net}_{15}', \text{Net}_{16}'$ für $k = 0 \succ \text{Net}_{13}', \text{Net}_{17}', \text{Net}_{16}'$ für $k = 1, 2, \dots$

Die Eigenschaft der Transitivität von \leq überträgt sich direkt auf $\succ_{\mu, \leq}^{\pi}$. Ist \leq total, überträgt sich auch diese Eigenschaft.

Korollar 145

Die Präferenz $\succ_{\mu, \leq}^{\pi}$ induziert eine Quasiordnung auf der Menge der gewichteten Netze über \mathbb{K} , für die π eine Perspektive ist. Ist \leq total, ist $\succ_{\mu, \leq}^{\pi}$ ebenfalls total.

Im Allgemeinen ist $\succ_{\mu, \leq}^{\pi}$ jedoch keine Halbordnung, da die Antisymmetrie nicht vorliegt: Es können durchaus zwei unterschiedliche Netze mit gleichen Kosten existieren. Daher geht auch die Existenz einer eindeutigen, kleinsten oberen Schranke für zwei Netze verloren. Betrachten wir jedoch anstatt der Netze selbst die Äquivalenzklassen bezüglich der durch $\succ_{\mu, \leq}^{\pi}$ induzierten Äquivalenzrelation, bilden diese wieder einen vollständigen Verband.

Manchmal möchten wir die Präferenz betrachten, die indifferent bezüglich jedem Paar zweier Netze ist. Diese bezeichnen wir als *Indifferenz*:

Notation 146 (Indifferenz)

Wir schreiben INDIFF für die Präferenz mit $\langle N, N' \rangle \in \text{INDIFF}$ für alle Netze N und N' .

3.8 Diskussion

In diesem Abschnitt vergleichen wir den entwickelten Formalismus der gewichteten Netze mit anderen Formalismen. Dazu begründen wir zunächst die Wahl des zugrunde liegenden Formalismus der Petrinetze und grenzen gewichtete Netze gegen andere Formalismen ab.

3.8.1 Petrinetze als Grundlage

Der grundlegende Formalismus für gewichtete Netze sind Petrinetze. Formale Modellierung mit Petrinetzen ist eine erprobte Technik zur Modellierung verteilter Systeme, unternehmensübergreifender Geschäftsprozesse und Services. Es existieren formale Petrinetzsemantiken für Industriesprachen wie BPEL [Jor+07] und BPMN [CT12] zur Modellierung offener Systeme sowie Compiler zur Übersetzung solcher Modelle in Petrinetze [Sta05; Loh08; DDO08a] und teilweise sogar zurück [LK08]. Zudem finden Petrinetze Anwendung in

Verwaltungssystemen für Geschäftsprozesse [Aal02] und sind das Ergebnis von Process-Mining-Techniken [Aal+12]. Petrinetze erlauben die Modellierung lokaler Zustände und Zustandsübergänge. Nebenläufigkeit wird explizit modelliert und von Nicht-Determinismus unterschieden. Daher eignen sich Petrinetze besonders gut für die Modellierung nebenläufiger Komponenten. *Prozessalgebren* [Bae05] beschreiben das Verhalten verteilter Systeme mit algebraischen Mitteln. Wie bei Petrinetzen liegt der Fokus auf expliziter Beschreibung von Nebenläufigkeit. Die Semantik eines prozessalgebraischen Terms kann als ein Transitionssystem, Automat oder Petrinetz aufgefasst werden. Der Fokus liegt in den klassischen Prozessalgebren CCS [Mil89], CSP [Hoa78] und ASP [BW90] auf der synchronen Kommunikation. Asynchrone Kommunikation kann dann über Pufferprozesse emuliert werden, die jedoch bei unbeschränkter Pufferung unendlich groß werden können. In der Literatur tauchen außerdem Prozessalgebren zur Modellierung asynchroner Kommunikation auf [BKP92; San01]. Wir haben uns für Petrinetze entschieden, da für unsere Zwecke nützliche Analysemethoden existieren.

Gewichtete Netze sind eng mit *gewichteten Automaten* [DKV09] verwandt, einer Generalisierung endlicher Automaten für die Modellierung quantitativer Sprachen. Der Zusammenhang zwischen Petrinetzen und formalen Sprachen ist ein gut erschlossenes Forschungsgebiet. Kurz gesagt können wir jeden endlichen Automaten als ein Petrinetz auffassen, daher können Petrinetze reguläre Sprachen erkennen. Diese von endlichen Automaten eingefangene Klasse der Petrinetze wird als *beschränkte* Petrinetze bezeichnet. Es gibt auch endliche, *unbeschränkte* Petrinetze, die dann einen unendlichen Zustandsraum aufweisen. Dies ist insbesondere bei der Modellierung asynchroner Interaktion durch Nachrichtenaustausch über Puffer von Vorteil (siehe auch Abschnitt 7.4.2, S. 162): Ein endliches Automatenmodell erfordert die Vorgabe von Puffergrößen; Petrinetze erlauben die Modellierung beliebiger Puffer. Da die Beschränktheit eines endlichen Petrinetzes entscheidbar ist, kann entschieden werden, ob ein komponiertes System einen unbeschränkten Puffer bräuchte oder nicht. Die existierenden Analyse- und Synthesetechniken für Petrinetze stammen aus dem Bereich verteilter Systeme, während vergleichbare Techniken im Bereich gewichteter Automaten sprachbasierte Probleme lösen.

Ein weiterer verwandter Formalismus sind *zeitbehaftete Modelle*, wie zum Beispiel *Zeitpetrinetze* [Pop13] und *Timed Automata* [AD94]: Modelle, die zeitliche Aspekte durch Annotationen in Form von Dauern, Uhren oder Schaltintervallen modellieren. Diese Modelle eignen sich insbesondere zur Modellierung zeitkritischer Anwendungen. Einerseits besteht ein offensichtlicher Zusammenhang zwischen Zeit und Kosten: Zeit kann als eine bestimmte Art von Kosten angesehen werden. Andererseits sind gewichtete Netze keine Verallgemeinerung zeitbehafteter Modelle: In gewichteten Netzen sind die Kosten rein nicht-funktionale Aspekte, die beispielsweise keinen Einfluss

auf das Eintreten eines Zustandsübergangs haben. Im Gegensatz dazu haben die Annotationen eines zeitbehafteten Systems direkten Einfluss auf die funktionalen Aspekte des Verhaltens: Ein Zustandsübergang kann beispielsweise nur in einem bestimmten Zeitintervall eintreten, wodurch beispielsweise Verklemmungen entstehen können. Insofern können wir gewichtete Netze und zeitbehaftete Systeme als orthogonale Aspekte verstehen. Für eine Klassifizierung der zeitbehafteten Formalismen im Kontext service-orientierter Architekturen verweisen wir auf die Übersichtsarbeit von Duske und Müller [DM12].

Außerdem betrachten Chothia und Kleijn [CK07] die Klasse der *Q-Automaten*. Diese verbinden wie gewichtete Netze die Betrachtung quantitativer Aspekte mit der Interaktion durch Komposition. Hier liegt der größte Unterschied im gewählten Kommunikationsmodell: Die Interaktion findet synchron statt. Beschränkte, asynchrone Kommunikation kann durch das Einfügen expliziter Pufferprozesse nachgebildet werden. Die Modellierung unbeschränkter Puffer ist jedoch auf endliche Weise nicht möglich.

Calvanese und Santoso [CS12] übertragen das *Roman Model* [Cal+08] auf *gewichtete Transitionssysteme*. Hier wird ebenfalls von synchroner Kommunikation ausgegangen. Insbesondere existieren zentrale, das Gesamtsystem steuernde Komponenten (*Orchestrators*), die Aufgaben an andere Systeme delegieren.

Zahlreiche Autoren wie beispielsweise Yu und Lin [YL05] sowie Zeng u. a. [Zen+04] modellieren *komplexe Services* in Form von Petrinetzen oder verwandter Modelle, in denen jede Transition einen *einfachen Service* repräsentiert. Es wird keine Interaktion zwischen den Services modelliert. Da das Schalten einer Transition instantan geschieht und stets dasselbe Verhalten (Konsum und Produktion von Marken) aufweist, haben solche einfachen Services keinen modellierten, lokalen Zustand.

3.8.2 Endliche, sequenzielle Abläufe als Grundlage für die Wertsemantik

In dieser Arbeit betrachten wir nur die endlichen, sequenziellen Abläufe eines Systems. Genauso wäre es möglich, andere bekannte Typen von Abläufen zu betrachten:

- Betrachtet man *halbgeordnete* anstatt sequenziellen Abläufen, sind die Schritte in einem Ablauf nicht mehr total sondern nur halbgeordnet. Die Verrechnung der entlang eines halbgeordneten Ablaufs auftretenden Kosten erfordert einen Aggregator, der Halbordnungen anstatt Folgen betrachtet. Eine kanonische Erweiterung der Begriffe in dieser Arbeit bestünde darin, halbgeordnete Kostenwerte durch die Aggregation aller

Linearisierungen zu verrechnen. Dann ist es jedoch unerheblich, ob die sequenziellen oder halbgeordneten Abläufe betrachtet werden.

- Die Betrachtung *unendlicher* Abläufe erfordert erneut eine Generalisierung der Begriffe in dieser Arbeit: Ein Kostenmodell definiert die Aggregation von Kostenwerten entlang endlicher Folgen. Für die Betrachtung unendlicher Abläufe müssten Operatoren betrachtet werden, die auch für unendliche Abläufe definiert sind. Dabei könnte man weitgehende Forschung auf die Arbeiten [CDH08; ÉK09; DM11] stützen.

Analog wird auch der Begriff der Perspektive komplexer.

Unsere Betrachtung beschränkt sich bezüglich Kosten auf das *Linear Time*-Verhalten [Pnu77] eines Systems und nicht auf sein *Branching Time*-Verhalten [BPM83]: Zwei Systeme mit der gleichen *Linear-Time Semantik* weisen in dieser Arbeit dieselbe Wertsemantik auf. Eine weitere mögliche Generalisierung bestünde in der Betrachtung einer Wertsemantik auf der Basis des *Branching-Verhaltens*. Eine Perspektive müsste dann Bäume von Schritten auf Bäume von Schritten abbilden. Ein Kostenmodell müsste dann Bäume von Kostenwerten anstatt Familien von Folgen aggregieren können. Hier können wir uns die Anwendung von Konzepten aus dem Forschungsgebiet der *gewichteten Baumatomen* [FV09] vorstellen.

3.8.3 Formalisierung von Perspektiven

Eine Perspektive bildet eine Familie von Transitionsfolgen auf eine neue Familie von Transitionsfolgen ab. Diese Definition ist sehr allgemein und wir beschränken uns in den Sätzen dieser Arbeit gleichzeitig auf nur wenige, ausgewählte Perspektiven. Die allgemeine formale Definition entstand als Versuch der Bildung eines kleinsten gemeinsamen, technisch leicht zu definierenden Nenners für die beschriebenen Perspektiven.

Wir betrachten in dieser Arbeit bewusst Familien von Folgen anstatt Mengen oder Multimengen, da mehrere und sogar unendlich viele Abläufe dieselben Folgen von Kostenwerten erzeugen können.

3.8.4 Formalisierung von Kostenmodellen

Ein Kostenmodell ist in dieser Arbeit eine Struktur mit zwei Funktionen \sum und \prod , wobei \sum Familien und \prod Folgen aggregiert. Ein häufig in dieser Arbeit betrachtete Spezialfall besteht darin, \sum und \prod durch zwei binäre Operationen $+$ (assoziativ und kommutativ) und \circ (assoziativ) zu charakterisieren. Diese Charakterisierung ist jedoch nicht immer möglich:

- Nicht jede assoziative und kommutative Operation $+$ ist auf unendlichen Familien definiert. Ein Netz hat jedoch im Allgemeinen unendlich viele Abläufe, die sogar paarweise unterschiedliche Kosten produzieren können. Wie in Abschnitt 2.5.2, S. 23 beschrieben, können wir zwar Idempotenz einzelner Elemente und die mögliche Existenz eines neutralen Elements nutzen, um $+$ auf einige unendliche Familien zu erweitern. Wir führen in dieser Arbeit außerdem eine kanonische Totalisierung eines semitotalen Kompressors durch Einführung eines Symbols ω ein (siehe Abschnitt 2.3.2, S. 17). Diese Totalisierung ist kanonisch – der entstehende Kompressor ist jedoch nur eine technische Darstellung einer partiellen als eine totale Funktion.
- Nicht alle gängigen Funktionen zur Aggregation, wie zum Beispiel der Durchschnitt oder die Discounted Sum lassen sich ohne weiteres durch binäre Operationen charakterisieren.

Bewertungs-Monoide und Multi-Hemiringe sind etwas spezieller als unser Begriff eines Kostenmodells. Beide Strukturen haben ihren Platz im Bereich der gewichteten Automaten und lassen die Modellierung interessanter Aggregationen wie dem Mittelwert oder der Discounted Sum zu [DK13]. Die Strukturen definieren jedoch im Allgemeinen nicht die Aggregation unendlich vieler Werte entlang einer Familie und die Totalisierung eines Bewertungsmonoids (Multi-Hemirings) ist im Allgemeinen nicht wieder ein Bewertungsmonoid (Multi-Hemiring).

Schließlich betrachten wir Familien anstatt Multimengen, da ein Element aus dem Träger in einer Multimenge nur endlich häufig vorkommen kann, es jedoch möglich ist, dass unendlich viele Abläufe denselben Kostenwert aufweisen.

3.8.5 Worst-Case- vs. Best-Case-Betrachtung

In dieser Arbeit betrachten wir häufig Verbands-Kostenmodelle. Wie bereits in Abschnitt 3.3.1, S. 49 beschrieben, modellieren solche Kostenmodelle den Worst Case, da die Kosten über die Supremum-Operation aggregiert werden. Auf den ersten Blick können wir analog den Best Case betrachten, indem wir den Verband umdrehen, sodass das Kostenmodell das Infimum anstatt des Supremums wählt. Dies zieht jedoch einen konzeptuellen Bruch nach sich: In Abschnitt 3.7, S. 91 führen wir die Präferenz *kostengünstiger* auf Basis der Ordnung des Kostenmodells ein. Dabei nehmen wir an, dass niedrige Kosten bezüglich der Ordnung gegenüber höheren Kosten bevorzugt werden. Drehen wir also die Ordnung des Kostenmodells schlicht um, bevorzugen wir mit einem Mal höhere Kosten gegenüber niedrigen. Zusammengefasst sind Kostenminimierung für den Worst Case und Kostenmaximierung für

den Best Case äquivalent. Betrachten wir die Anforderung der k -Kostenbeschränktheit aus Abschnitt 3.6, S. 86, beobachten wir zudem einen weiteren konzeptuellen Unterschied zwischen dem Worst Case und dem Best Case: Betrachten wir die Kosten des Worst Cases, erkennen wir die *Verletzung* der Kostenbeschränktheit an einem einzigen Ablauf mit Kosten $k' \not\leq k$. Also können wir k -Kostenbeschränktheit als Sicherheitseigenschaft einstufen. Fordern wir dagegen k -Kostenbeschränktheit im Best Case, erkennen wir die *Erfüllung* der Kostenschranke an jedem Ablauf mit Kosten $\leq k$. Daher lässt sich k -Kostenbeschränktheit im Best Case eher als ein Erreichbarkeitsproblem einstufen. Daher erfordert die Betrachtung des Best Cases im Allgemeinen andere Methoden als die, die in dieser Arbeit betrachtet werden.

Teil II

Analyse geschlossener Systeme

Die Analysefragen in diesem Teil betreffen die Wert-Semantik (siehe Definition 85, S. 67) eines gewichteten Netzes N über \mathbb{K} aus einer Perspektive π bezüglich eines Kostenmodells μ . Wir betrachten die folgenden Berechnungs- und Entscheidungsprobleme:

Wert-Semantik. Berechne die Wert-Semantik $\llbracket N \rrbracket_{\mu}^{\pi}$ von N .

k -Kostenbeschränktheit. Entscheide für ein gegebenes $k \in \mathbb{K}$ und eine Relation \leq auf \mathbb{K} die k -Kostenbeschränktheit von N .

Kostenbeschränktheit. Entscheide die Kostenbeschränktheit von N .

Bei den Entscheidungsproblemen der k -Kostenbeschränktheit und Kostenbeschränktheit nehmen wir wie in Definition 139, S. 90 an, dass μ ein Verbands-Kostenmodell ist.

Zur Lösung der Probleme wenden wir die Methodik der *Repräsentanten* (siehe Abschnitt 1.5, S. 12) an: Ein Repräsentant ist ein besonderes Netz, das sich besonders leicht analysieren lässt. Dabei ist der Begriff des Repräsentanten stets entsprechend des gerade betrachteten Problems parametrisiert. Wir betrachten in Kapitel 4, S. 101 unterschiedliche Formen von Repräsentanten und zeigen Konstruktionen vor, die ein gegebenes Netz N in einen äquivalenten Repräsentanten N' überführen. Die dabei gewonnenen Ergebnisse wenden wir in Kapitel 5, S. 117 an, um die obigen Probleme zu lösen.

4 Repräsentanten geschlossener Systeme

In diesem Kapitel beschreiben wir unseren Lösungsansatz zur Analyse geschlossener Systeme mit *Repräsentanten* in Abschnitt 1.5, S. 12. Dabei verfolgen wir zwei Ziele: Erstens die weitmögliche Unabhängigkeit von der Perspektive in Abschnitt 4.1 und zweitens die Anwendbarkeit bestehender Analysemethoden in Abschnitt 4.2, S. 109 und Abschnitt 4.3, S. 113. In Abschnitt 4.4, S. 115 diskutieren wir die Ergebnisse dieses Kapitels.

4.1 Perspektiven-Repräsentanten

In diesem Abschnitt reduzieren wir Probleme für eine Perspektive π auf Probleme für eine andere Perspektive π' . Dazu beobachten wir, dass in einigen Netzen die Kosten für ein Kostenmodell μ aus der Perspektive π mit den Kosten aus der Perspektive π' zusammenfallen. Solch ein Netz nennen wir einen π, π' -*Perspektiven-Repräsentanten* bezüglich μ .

Perspektiven-
Repräsentanten

Definition 147 (Perspektiven-Repräsentant)

Sei N ein gewichtetes Netz über \mathbb{K} , π und π' Perspektiven von N und μ ein Kostenmodell über \mathbb{K} . Falls $\llbracket N \rrbracket_{\mu}^{\pi} = \llbracket N \rrbracket_{\mu}^{\pi'}$, ist N ein π, π' -*Perspektiven-Repräsentant* bezüglich μ .

Beispiel 148

Wir betrachten das Netz Net_{16} aus Abbildung 16, S. 68 aus den Perspektiven $\pi = \text{FILTER}[t_1, t_2, t_3]$ und $\pi' = \text{TERM}[\text{Net}_{13}] \triangleright \pi$. Wählen wir als Kostenmodell beispielsweise die Max-Plus-Algebra, stellen wir fest, dass die Kosten aus beiden Perspektiven zusammenfallen: $\llbracket \text{Net}_{13} \rrbracket_{\text{MP}}^{\pi} = \llbracket \text{Net}_{13} \rrbracket_{\text{MP}}^{\pi'} = k$. Daher ist Net_{16} ein π, π' -Repräsentant bezüglich MP.

Betrachten wir dagegen das Netz Net_{14} aus Abbildung 14, S. 44, beobachten wir $\llbracket \text{Net}_{14} \rrbracket_{\text{MP}}^{\pi} = 1 \neq 0 = \llbracket \text{Net}_{14} \rrbracket_{\text{MP}}^{\pi'}$.

Wir merken an, dass jeder π, π' -Perspektiven-Repräsentant auch ein π', π -Perspektiven-Repräsentant ist. Die Reihenfolge der Perspektiven π, π' spielt genau dann eine Rolle, wenn wir nach einem *äquivalenten* π, π' -Perspektiven-Repräsentanten zu einem gegebenen Netz suchen.

Definition 149 (Äquivalenter Perspektiven-Repräsentant)

Seien N, N' gewichtete Netze über \mathbb{K} . Sei μ ein Kostenmodell über \mathbb{K} . Seien π, π' Perspektiven von N und N' , sodass

- (1) $\llbracket N \rrbracket_\mu^\pi = \llbracket N' \rrbracket_\mu^\pi$ und
- (2) N' ist ein π, π' -Perspektiven-Repräsentant.

Dann ist N' ein zu N äquivalenter π, π' -Perspektiven-Repräsentant.

Aus den beiden Forderungen können wir folgern, dass $\llbracket N \rrbracket_\mu^\pi = \llbracket N' \rrbracket_\mu^\pi = \llbracket N' \rrbracket_\mu^{\pi'}$ gilt. Im Allgemeinen unterscheiden sich die Kosten von N und N' jedoch aus der Perspektive π' ; es kann also $\llbracket N \rrbracket_\mu^{\pi'} \neq \llbracket N' \rrbracket_\mu^{\pi'}$ gelten.

Hat ein Netz N aus beiden Perspektiven π und π' dieselben Familien von Kostensequenzen, dann ist N stets ein π, π' -Perspektiven-Repräsentant – unabhängig von der konkreten Wahl des Kostenmodells.

Lemma 150

Sei N ein gewichtetes Netz über \mathbb{K} , π und π' Perspektiven von N und μ ein Kostenmodell über \mathbb{K} . Falls $\llbracket \pi(N) \rrbracket_N = \llbracket \pi'(N) \rrbracket_N$ gilt, ist N ein π, π' -Perspektiven-Repräsentant bezüglich μ .

Beweis (Lemma 150)

Aus $(\llbracket w \rrbracket_N)_{w \in \pi(N)} = (\llbracket w \rrbracket_N)_{w \in \pi'(N)}$ folgern wir $\mu((\llbracket w \rrbracket_N)_{w \in \pi(N)}) = \mu((\llbracket w \rrbracket_N)_{w \in \pi'(N)})$, woraus direkt $\llbracket N \rrbracket_\mu^\pi = \llbracket N \rrbracket_\mu^{\pi'}$ folgt. Nach Definition 147, S. 101 ist N dann ein π, π' -Perspektiven-Repräsentant bezüglich μ .

In den folgenden Unterabschnitten stellen wir Verfahren zur Konstruktion eines äquivalenten Perspektiven-Repräsentanten aus einem Ausgangsnetz für unterschiedliche Paare π, π' von Perspektiven aus Notation 42, S. 43 auf. Mit *äquivalent* meinen wir dabei die Gleichheit der Kosten aus der Perspektive π . Bei einigen Konstruktionen fordern wir bestimmte Eigenschaften des Kostenmodells oder des Ausgangsnetzes, bei anderen können wir allgemeinere Aussagen treffen. Notation 42, S. 43. Abbildung 21, Abbildung 22 und Abbildung 23 fassen die Ergebnisse zusammen: Eine Kante von A nach B in Abbildung 21 bedeutet, dass ein äquivalenter A, B -Perspektiven-Repräsentant für beliebige Netze und Kostenmodelle existiert.

In Abbildung 22 beschreibt eine blaue (rote) Kante von A nach B die Existenz eines A, B -Perspektiven-Repräsentanten für beliebige Netze bezüglich eines Verbands-Kostenmodells (der Max-Plus-Algebra als Kostenmodell).

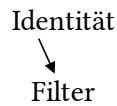


Abbildung 21 Existenz äquivalenter Perspektiven-Repräsentanten



Abbildung 22 Existenz äquivalenter Perspektiven-Repräsentanten, teils nur für Verbands-Kostenmodelle (blau) oder der Max-Plus-Algebra als Kostenmodell (rot)

Abbildung 23 zeigt zusätzliche, konditionale Resultate: Eine gestrichelte Kante von A nach B bezeugt die Existenz eines äquivalenten A, B -Perspektiven-Repräsentanten sofern das Ausgangsnetz eine bestimmte Eigenschaft erfüllt.

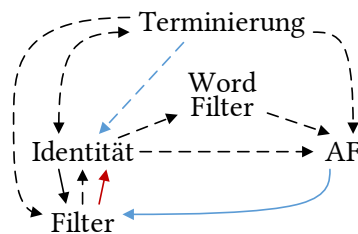


Abbildung 23 Konditionale Existenz äquivalenter Perspektiven-Repräsentanten (gestrichelt), teils nur für Verbands-Kostenmodelle (blau) oder die Max-Plus-Algebra als Kostenmodell (rot)

Existenz

4.1.1 Von der Identität zu Filter-Perspektiven

Sei N ein Netz mit Transitionen T . Dann ist N offenbar ein $\text{ID}, \text{FILTER}[T]$ -Perspektiven-Repräsentant bezüglich beliebiger Kostenmodelle. Es existiert also eine Filter-Perspektive π , sodass N ein ID, π -Perspektiven-Repräsentant ist.

Nehmen wir jedoch eine konkrete Filter-Perspektive $\pi' = \text{FILTER}[T']$ mit $T' \subset T$ als gegeben an. Dann ist N im Allgemeinen kein ID, π' -Perspektiven-Repräsentant. Abhängig vom Kostenmodell kann es möglich sein, Kosten von einer Transition $t \notin T'$ zu einer anderen Transition $t' \in T'$ zu verlagern. Wenn

dies für alle $t \in T \setminus T'$ möglich ist, können wir N in einen id, π' -Perspektiven-Repräsentanten transformieren.

4.1.2 Wechsel zwischen Identität und Terminierungs-Perspektive

Sei N ein Netz, in dem jede Markierung eine Zielmarkierung ist. Dann ist N sowohl ein $\text{id}, \text{TERM}[N]$ -Perspektiven-Repräsentant wie ein $\text{TERM}[N], \text{id}$ -Perspektiven-Repräsentant. Die Umwandlung eines beliebigen Netzes in einen äquivalenten $\text{id}, \text{TERM}[N]$ -Perspektiven-Repräsentanten oder einen äquivalenten $\text{TERM}[N], \text{id}$ -Perspektiven-Repräsentanten lässt sich also durch Änderung der Zielmarkierungen erreichen. Die Anpassung der Zielmarkierungen zieht allerdings in unserem Kontext weitreichende Änderungen mit: So kann beispielsweise das Netz durch Anpassung der Zielmarkierungen schwach terminierend werden.

Betrachten wir ein bezüglich N monotones Verbands-Kostenmodell μ über \mathbb{K} , können wir genauere Aussagen treffen. Ist N schwach terminierend, lassen sich alle Abläufe zu terminierenden Abläufen verlängern. Aus der Monotonie können wir folgern, dass diese Verlängerung entweder in gleichen oder höheren Kosten resultiert. Daher ist N sowohl ein $\text{id}, \text{TERM}[N]$ -Perspektiven-Repräsentant wie ein $\text{TERM}[N], \text{id}$ -Perspektiven-Repräsentant.

4.1.3 Von der Terminierungs-Perspektive zur Filter-Perspektive

Sei N ein Netz mit Transitionen T . Wir haben bereits in Abschnitt 4.1.2 die Existenz äquivalenter $\text{TERM}[N], \text{id}$ -Perspektiven-Repräsentanten diskutiert. Dabei war entweder eine Anpassung der Zielmarkierungen oder sowohl die schwache Terminierung des Netzes sowie ein besonders beschaffenes Kostenmodell eine Voraussetzung unserer Betrachtungen. Da wir id als eine spezielle Filter-Perspektive betrachten können (siehe Abschnitt 4.1.1, S. 103), ist unter diesen Umständen N auch ein $\text{TERM}[N], \text{FILTER}[T]$ -Perspektiven-Repräsentant.

Möchten wir jedoch direkt von der Terminierungs-Perspektive zur Filter-Perspektive springen, können wir die Anforderungen an die Menge der Zielmarkierungen bzw. der schwachen Terminierung von N auflockern. Konkret fordern wir dann einzig die Abgeschlossenheit der Menge M von Zielmarkierungen von N bezüglich Überdeckung, das heißt für alle Markierungen m, m' mit $m \sqsubseteq m'$ gelte: $m \in M$ impliziert $m' \in M$. Die Idee hinter dieser Forderung ist die folgende: Ist M abgeschlossen bezüglich Überdeckung, können wir für jede Zielmarkierung $m \in M$ eine Transition t_m einfügen, die genau dann feuern kann, wenn N in m ist. Existierte wenigstens eine erreichbare Markierung $m' \notin M$ mit $m \sqsubseteq m'$ für wenigstens eine Markierung $m \in M$, dann können wir

keine Transition dieser Art einfügen, da eine Transition, die in m aktiviert ist, auch in m' aktiviert ist. Die Menge $T_M = \{t_m \mid m \in M\}$ spielt in der folgenden Konstruktion eine entscheidende Rolle.

Wir betrachten ein beliebiges Verbands-Kostenmodell μ über \mathbb{K} . Sei G ein Graph der Menge $\mathbb{K}[N]$ in μ . Intuitiv repräsentieren wir jeden terminierenden Ablauf w durch einen Ablauf wt_m und ignorieren die Kosten der Transitionen aus w . Dazu wenden wir zunächst die Konstruktion aus Abschnitt 3.5.1.2, S. 75 für die Perspektive id an. Das resultierende Netz $\text{REP}_\mu^\pi(N, G)$ ist kosteneindeutig und die Kosten eines Ablaufs lassen sich aus dem zusätzlichen, markierten Platz ablesen. Für jeden dieser Plätze p und $t_m \in T_M$ sei t_m^p eine Transition mit Kosten p . Entstehe N' aus $\text{REP}_\mu^\pi(N, G)$ wie folgt:

- (1) Füge einen initial markierten Platz \hat{p}_{ctrl} ein, der in allen Rahmenmarkierungen und Zielmarkierungen markiert und in keiner Rahmenmarkierung und keiner Zielmarkierung unmarkiert vorkommt.
- (2) Füge alle t_m^p so ein, dass t_m^p eine Schlinge mit Gewicht $m(p')$ für alle Plätze p' von N hat, eine Schlinge mit Gewicht 1 zu p hat, und außerdem eine Marke von \hat{p}_{ctrl} konsumiert.

Im resultierenden Netz N' sind also alle Abläufe aus $\text{REP}_\mu^\pi(N, G)$ und damit auch aus N möglich. Außerdem kommt für jeden terminierenden Ablauf w der Ablauf wt_m^p hinzu. Unter dieser Veränderung bleibt die Erfüllung bzw. Nicht-Erfüllung der schwachen Terminierung erhalten, da das Eintreten von t_m^p einzig eine Marke vom Platz \hat{p}_{ctrl} entfernt. Wir beobachten außerdem, dass $\mu(\llbracket w \rrbracket_N) = \mu(\llbracket t_m^p \rrbracket_{N'})$ gilt. Sei T' die Menge aller Transitionen t_m^p . Daher ist N' ein $\text{TERM}[N']$, $\text{FILTER}[T']$ -Perspektiven-Repräsentant: Filtern wir alle ursprünglichen Transitionen heraus, wird aus jedem Ablauf wt_m^p der Ablauf t_m^p .

Wir betrachten die Frage, wann N' ein endliches Netz ist. Betrachten wir zunächst $\text{REP}_\mu^\pi(N, G)$, können wir die Bedingungen Korollar 106, S. 77 entnehmen. Da wir nur \hat{p}_{ctrl} sowie die Transitionen aus T' einfügen, ist N' endlich, falls $\text{REP}_\mu^\pi(N, G)$ und T' endlich sind. Die Endlichkeit von T' hängt wiederum von der Endlichkeit von T_M ab. Wir können T_M genau dann endlich wählen, wenn N nur endlich viele Plätze hat und M endlich erzeugt ist, das heißt, eine endliche Menge $M' \subseteq M$ existiert, sodass für alle $m \in M$ ein $m' \in M'$ existiert mit $m' \subseteq m$. Zusammenfassend können wir schließen: N' ist genau dann endlich, wenn die folgenden Bedingungen alle erfüllt sind:

- (1) N ist endlich.
- (2) G ist endlich.
- (3) M ist endlich erzeugt.

Ist μ assoziativ, können wir immer $\text{REPAssoc}_\mu^\pi(N)$ betrachten und die zweite Bedingung ersetzen durch: $(\mathbb{K}[N])^\mu$ ist endlich.

Da der Platz \hat{p}_{ctrl} in keiner Rahmenmarkierung unmarkiert ist, ist wt_m^p kein terminierender Ablauf. Daher bewahrt die Konstruktion schwache Terminierung *nicht*. Basiert μ auf einem Semiring, können wir den Transitionen, die nicht in T' enthalten sind, Kosten $\mathbb{1}$ zuweisen. Wenn wir anschließend die Rahmenmarkierungen und Zielmarkierungen von N' so anpassen, dass \hat{p}_{ctrl} sowohl markiert wie unmarkiert enthalten ist, erhalten wir einen äquivalenten $\text{TERM}[N']$, $\text{FILTER}[T']$ -Repräsentanten, der schwach terminierend ist genau dann wenn N schwach terminierend ist. Das heißt, in diesem Falle bewahrt die Konstruktion schwache Terminierung.

Anstatt $\text{REP}_\mu^\pi(N, G)$ wie in Abschnitt 3.5.1.2, S. 75 zu konstruieren, können wir gegebenenfalls auch die Konstruktionen aus Abschnitt 3.5.1.3, S. 78 oder Abschnitt 3.5.1.4, S. 79 verwenden. Entscheiden wir uns für die Konstruktion aus Abschnitt 3.5.1.3, S. 78, fügen wir anstatt t_m^p für jedes $m \in M$ nur eine Transition t_m ein, die ein Subnetz aktiviert, indem Transitionen für jeden Kostenwert aus N vorliegen. An dieser Stelle müssen wir jedoch zusätzlich Monotonie des Kostenmodells fordern. Folgen wir der Konstruktion aus Abschnitt 3.5.1.4, S. 79, besteht dieses Subnetz nur aus einer Transition t mit Kosten 1, die durch jede Transition t_m aktiviert wird und die Kosten aus \hat{p} konsumiert. Dafür benötigen wir gegebenenfalls zusätzlich eine Transition mit Kosten $-\infty$ mit einer Schlinge zu $\hat{p}_{-\infty}$, die alternativ zu t eintreten kann. In beiden Fällen fertigen wir zusätzlich eine Kopie von $\text{REP}_\mu^\pi(N, G)$ an, und t_m^p wechselt von Original zu Kopie, damit keine Kosten mitgezählt werden, die nach dem Schalten von t_m^p anfallen. Falls wir eine dieser beiden Konstruktionen anwenden können, ist N' genau dann endlich, wenn N endlich ist und M endlich erzeugt ist (siehe Korollar 110, S. 78 und Korollar 120, S. 83).

4.1.4 Von der Identitäts-Perspektive zur Anwendungsfall-Perspektive

Sei N ein beschriftetes Netz über Σ . Kommt ε nicht als Beschriftung vor, können wir stets den Anwendungsfall U bilden mit $\bullet U = \{\varepsilon\}$ und $U^\bullet = \Sigma^*$. Dann gilt: N ist ein ID , $\text{AF}[N, U]$ -Perspektiven-Repräsentant.

Ist ein bestimmter Anwendungsfall U gegeben, ist es nicht ohne weitere Annahmen möglich, einen ID , $\text{AF}[N, U]$ -Perspektiven-Repräsentanten zu konstruieren, aus den gleichen Gründen wie in Abschnitt 4.1.1, S. 103.

4.1.5 Von Anwendungsfall-Perspektive zur Filter-Perspektive

Sei N ein gewichtetes Netz und U ein Anwendungsfall. Dann teilt U die Abläufe von N in zwei Mengen auf: Das U -Verhalten von N und die Menge der

restlichen Abläufe. Hier können wir eine Ähnlichkeit zwischen Anwendungsfälle und Zielmarkierungen beobachten. Genau wie U teilen die Zielmarkierungen M von N in die Mengen der terminierenden und der restlichen Abläufe. Trotzdem unterscheiden sich Anwendungsfälle und Zielmarkierungen in den folgenden Punkten:

- (1) Zielmarkierungen klassifizieren Abläufe auf Basis ihrer resultierenden Markierung. Bei Anwendungsfälle ist dies nicht der Fall: Seien v, w zwei Abläufe, die in derselben Markierung resultieren. Dann ist es möglich, dass v im U -Verhalten enthalten ist, während w nicht im U -Verhalten enthalten ist.
- (2) Ein Anwendungsfall U teilt jeden der Abläufe aus dem U -Verhalten in zwei Teile: Den Präfix, der in $\bullet U$ enthalten ist und den Suffix, der in $U\bullet$ enthalten ist. Diese Aufteilung ist für einen Ablauf w nicht eindeutig: Es können mehrere Präfixe v von w existieren mit $v \in \bullet U$.

Aus dem zweiten Punkt ergibt sich auch der Unterschied zwischen der Terminierungs-Perspektive und der Anwendungsfall-Perspektive: Während die Terminierungs-Perspektive jeden Ablauf genau einmal zurück gibt, kann die Anwendungsfall-Perspektive denselben Ablauf mehrfach zurückgeben, nämlich für jede von U induzierte Aufteilung in Präfix und Suffix.

Trotzdem lassen sich die Ergebnisse aus Abschnitt 4.1.3, S. 104 verwenden, um einen $\text{AF}[U], \text{FILTER}[T']$ -Perspektiven-Repräsentanten aus N und U zu konstruieren. Intuitiv kodieren wir zunächst U in N und setzen die Zielmarkierungen, sodass wir direkt die Konstruktion aus Abschnitt 4.1.3, S. 104 anwenden können.

Seien dazu $\bullet A$ und $A\bullet$ Automaten, die $\bullet U$ und $U\bullet$ akzeptieren. Ohne Beschränkung der Allgemeinheit seien $\bullet A$ und $A\bullet$ total bezüglich aller Beschriftungen in N . Habe N die Transitionen Z , $\bullet A$ die Transitionen $T_{\bullet A}$ und $A\bullet$ die Transitionen $T_{A\bullet}$ und die Endzustände Q . Wir bilden das Netz N_U aus N und U wie folgt: Synchronisiere jede Transition $t \in T$ mit jeder gleich beschrifteten Transition aus $t' \in V \cup W$ (siehe Abschnitt 2.7.7, S. 35). Jede resultierende Transition stammt also aus $T \times V$ oder $T \times W$. Sei M_Q die Menge der Markierungen, in denen wenigstens ein Platz aus Q markiert ist. Sei μ ein Verbands-Kostenmodell über \mathbb{K} und G ein Graph der Menge $\mathbb{K}[N]$ in μ . Wir können jetzt die Konstruktion aus Abschnitt 4.1.3, S. 104 anwenden, wobei wir M_Q anstatt der Menge der Zielmarkierungen verwenden. Wir beschreiben die Konstruktion noch einmal gesondert.

Dazu wenden wir zunächst die Konstruktion aus Abschnitt 3.5.1.2, S. 75 für die Perspektive $\pi' = \text{FILTER}[T \times W]$ auf N_U an, die im Netz $\text{REP}_{\mu}^{\pi'}(N, G)$ resultiere. Anschließend konstruieren wir N'' aus $\text{REP}_{\mu}^{\pi'}(N, G)$ wie folgt.

- (1) Wir fügen einen initial markierten Platz \hat{p}_{ctrl} ein.

- (2) Sei $q \in Q$. Sei p ein Platz von $\text{REP}_\mu^{\pi'}(N, G)$ mit $p \in \mathbb{K}$. Dann fügen wir eine Transition t_q^p mit Kosten p ein, die eine Marke von \hat{p}_{ctrl} konsumiert und eine Schlinge zu q und p hat.

Als Beschriftung für die neu eingefügten Transitionen T' können wir ein beliebiges $x \neq \varepsilon$ wählen, das nicht in N bereits vorkommt.

Das Netz N'' ist ein $\text{AF}[U], \text{FILTER}[T']$ -Perspektiven-Repräsentant: Eine Transition aus t_q^p wird genau durch Abläufe uv aktiviert, wobei $u \in \bullet U$, $v \in v \bullet$ und $\mu(v) = p$. Das Verfahren verändert weder Rahmenmarkierungen noch Zielmarkierungen, die hinzugefügten Transitionen ändern das restliche Verhalten nicht; damit wird schwache Terminierung bewahrt.

Die Endlichkeit des Netzes N'' hängt von der Endlichkeit von $\text{REP}_\mu^{\pi'}(N, G)$ ab. $\text{REP}_\mu^{\pi'}(N, G)$ ist laut Korollar 106, S. 77 genau dann endlich, wenn N_U und $(\llbracket T \times W \rrbracket_N)^\mu$ endlich ist. N_U ist genau dann endlich, wenn sowohl N wie auch die Automaten $\bullet A$ und $A \bullet$ endlich sind.

Wie in Abschnitt 4.1.3, S. 104 beschrieben, können wir auch hier anstatt der Konstruktion aus Abschnitt 3.5.1.2, S. 75 die entsprechenden Konstruktionen aus Abschnitt 3.5.1.3, S. 78 oder Abschnitt 3.5.1.4, S. 79 verwenden, falls die für die Konstruktionen notwendigen Voraussetzungen erfüllt sind. Die Auswirkungen sind dieselben wie in Abschnitt 4.1.3, S. 104.

4.1.6 Von der Identitäts-Perspektive zur Wort-Filter-Perspektive

Ist ein Netz N unbeschriftet, ist N ein $\text{ID}, \text{WFILTER}[N, \varepsilon]$ -Perspektiven-Repräsentant. Wir können also zu einem beliebigen Netz N immer die unbeschriftete Variante N' bilden. Dann ist N' ein zu N äquivalenter $\text{ID}, \text{WFILTER}[N, \varepsilon]$ -Perspektiven-Repräsentant.

Ist ein bestimmtes Wort w gegeben, ist es nicht ohne weitere Annahmen möglich, einen $\text{ID}, \text{WFILTER}[N, w]$ -Perspektiven-Repräsentanten zu konstruieren, aus den gleichen Gründen wie in Abschnitt 4.1.1, S. 103.

4.1.7 Von der Filter-Perspektive zur Identitäts-Perspektive

Sei N ein Netz mit Transitionen T . Sei $T' \subseteq T$. Dann ist N ein $\text{FILTER}[T']$, ID -Perspektiven-Repräsentant g.d.w. jedes $t \in T \setminus T'$ niemals schaltet. Dies ist trivial erfüllt falls $T' = T$.

Ist das Kostenmodell aus einem Semiring entstanden, können wir die Kosten jeder heraus gefilterten Transition $t \in T \setminus T'$ auf die Kosten des $\mathbb{1}$ -Elements setzen. Im Falle der Max-Plus-Algebra entspricht dies der 0.

4.1.8 Von der Wort-Filter-Perspektive zur Anwendungsfall-Perspektive

Sei N ein Netz und w ein Wort über Beschriftungen von N . Dann können wir zu w kanonisch den Anwendungsfall U bilden mit $\bullet U = \{\varepsilon\}$ und $U^\bullet = \{w\}$. Dann ist N ein $\text{WFILTER}[N, w]$, $\text{AF}[N, U]$ -Perspektiven-Repräsentant, falls wenigstens eine der beiden Bedingungen erfüllt ist:

- (1) In N liegen keine mit ε beschrifteten Transitionen vor. Dann entspricht jeder Ablauf, der w akzeptiert, einem Ablauf, der U überdeckt und anders herum. Existieren ε -Transitionen, gibt es folgenden Unterschied: Sei $w = uv$ ein Ablauf, wobei u und v Wörter von Transitionen sind und u nur aus ε -Transitionen besteht. Dann zählen wir w *mehrfach*, wenn wir die Anwendungsfall-Perspektive zugrunde legen: Für jeden Suffix u' von u zählen wir $u'v$. Im Falle der Wortfilter-Perspektive zählen wir den Ablauf w jedoch genau einmal.
- (2) Das Kostenmodell μ unterscheidet nicht zwischen zwei Familien $(a_i)_{i \in I}$ und $(b_j)_{j \in J}$, falls $\{a_i \mid i \in I\} = \{b_j \mid j \in J\}$, und für alle Abläufe w gilt: Die Kosten von w sind gleich der Kosten der Projektion von w auf nicht- ε -Transitionen. Dann hat die in Bedingung 1 beschriebene Abweichung der Abläufe keinen Einfluss auf die Kosten.

4.1.9 Von der Terminierungs-Perspektive zur Anwendungsfall-Perspektive

Sei N ein Netz mit Zielmarkierungen M^\diamond . Habe jede Transition t die Beschriftung t . Sei U der Anwendungsfall von N mit $\bullet U = \{\varepsilon\}$ und

$$U^\bullet = \bigcup_{m \in M^\diamond} \text{RUNS}(N, m) .$$

Dann ist N ein $\text{TERM}[N]$, $\text{AF}[U]$ -Perspektiven-Repräsentant von N .

Indem wir die Beschriftungen eines beliebigen Netzes N anpassen, können wir also kanonisch den passenden Anwendungsfall bilden, der genau die terminierenden Abläufe erfasst. Ohne die Änderung der Beschriftungen ist dies im Allgemeinen nicht möglich, da wir ansonsten möglicherweise nicht zwischen einem terminierendem und einem nicht-terminierenden Ablauf unterscheiden können.

4.2 Wert-Repräsentanten

In diesem Abschnitt betrachten wir das Problem, die Wertsemantik eines endlichen Netzes N zu berechnen. Der naive Ansatz zur Berechnung durch Betrachtung der Kosten jedes einzelnen Ablaufs ist nicht praktikabel: Die Menge

der Abläufe von N ist im Allgemeinen unendlich. Wir betrachten im Folgenden zunächst die Einflussfaktoren für die Berechenbarkeit der Wertsemantik. Anschließend schlagen wir die Analyse mithilfe von *Wert-Repräsentanten* vor.

Kompositionalität von Perspektive und Kostenmodell. Sind Perspektive und Kostenmodell kompositional (siehe Definition 45, S. 45 und Definition 59, S. 51), können wir das *Teile und Herrsche*-Prinzip anwenden: Wir können die Wertsemantik *portionsweise* berechnen. Insbesondere ist es dann möglich, die Kosten jeder erreichbaren Markierung anstatt der Kosten jedes Ablaufs zu betrachten (siehe Satz 93, S. 71). Wir nehmen daher im Folgenden kompositionale Perspektiven und Kostenmodelle an.

Kosteneindeutigkeit. Ist N kosteneindeutig (siehe Definition 95, S. 72), können wir die Kosten einer einzigen erreichbaren Markierung m aus den Kosten eines einzelnen in m resultierenden Ablaufs ableiten.

Beschränktheit vs. Unbeschränktheit. Ist N beschränkt, sind nur endlich viele Markierungen erreichbar. Ist N kosteneindeutig und beschränkt, genügt es, zur Berechnung der Wertsemantik nur endlich viele Abläufe zu betrachten. Ist N jedoch unbeschränkt, sind unendlich viele Markierungen erreichbar. Ist N P -kosteneindeutig und ist jeder Platz $p \in P$ beschränkt, ist die Menge der charakteristischen Markierungen endlich; das heißt, es genügt die Markierungen in Bezug auf die Plätze in P zu betrachten. Ist jedoch wenigstens ein Platz $p \in P$ unbeschränkt, ist die Bestimmung der Kosten erneut problematisch: Dann existieren nämlich unendlich viele erreichbare Markierungen, die sich durch die Anzahl der Marken auf einem Platz $p \in P$ unterscheiden.

Wert-Repräsentanten. Wie im vorherigen Absatz beschrieben, ist die P -Kosteneindeutigkeit eines Netzes von Nutzen in der Analyse, vor allem, wenn alle Plätze $p \in P$ beschränkt sind. Unter bestimmten Umständen kann die Analyse trotz Unbeschränktheit einer Menge $P' \subseteq P$ möglich sein, nämlich immer dann, wenn die Unbeschränktheit von P' stets mit einem bestimmten, bekannten Kostenwert zusammenfällt. Ein P -kosteneindeutiges Netz N , in dem für alle nicht leeren Teilmengen von P ein genauer Kostenwert *bekannt* ist, nennen wir einen Wert-Repräsentanten. Wir formalisieren *bekannt*, indem wir den Begriff des Wert-Repräsentanten mit einer Abbildung f parametrisieren:

Definition 151 (Wert-Repräsentant)

Sei N ein gewichtetes Netz über \mathbb{K} , μ ein kompositionales Kostenmodell über \mathbb{K} und π eine kompositionale Perspektive. Sei P eine Teilmenge der Plätze von N und $f: \wp(P) \setminus \emptyset \rightarrow \mathbb{K}$.

N ist ein f -Wert-Repräsentant, falls N jede der folgenden Eigenschaften aufweist:

- (1) N ist P -kosteneindeutig bezüglich π und μ .
- (2) Sei P' die Menge der in N unbeschränkten Plätze aus P . Dann ist P' entweder leer oder $f(P') = \llbracket N \rrbracket_{\mu}^{\pi}$.

Existenz äquivalenter Wert-Repräsentanten. Existiert wenigstens ein P -kosteneindeutiges Netz N mit Kosten x , existiert auch ein zu N kostenäquivalenter Wert-Repräsentant: Dazu wählen wir f als die konstante Abbildung $P' \mapsto x$. Im Szenario der Analyse ist jedoch x ausdrücklich nicht bekannt, sondern zu ermitteln.

Wir suchen im Folgenden nach Konstruktionen der folgenden Art: Gegeben sei ein Netz N über \mathbb{K} . Gesucht wird eine Funktion f und ein zu N kostenäquivalenter f -Wert-Repräsentant N' . Dabei nehmen wir an, dass N endlich ist und streben ebenfalls ein endliches N' an.

Wir zeigen in den folgenden Abschnitten die Existenz eines endlichen f -Wert-Repräsentanten für ein gegebenes und ein endliches Netz N , das durch die *Max-Plus-Algebra* induzierte Kostenmodell, sowie eine Filter-Perspektive. Anschließend zeigen wir, wie wir die Resultate aus Abschnitt 4.1, S. 101 anwenden können, um Wert-Repräsentanten für weitere Perspektiven zu konstruieren.

4.2.1 Konstruktion von Wert-Repräsentanten für Verbände

Sei N ein endliches, gewichtetes Netz über \mathbb{K} . Sei $\pi = \text{FILTER}[T]$ eine Filter-Perspektive von N und μ ein Verbands-Kostenmodell über \mathbb{K} . Sei G ein Graph von $\mathbb{K}[N, \pi]$ in μ .

Dann können wir zu N ein kostenäquivalentes (siehe Lemma 105, S. 76), Q -kosteneindeutiges (siehe Lemma 103, S. 75) Netz $\text{REP}_{\mu}^{\pi}(N, G)$ konstruieren (siehe Abschnitt 3.5.1.2, S. 75).

Jede Transition, die Marken auf einen Platz $q \in Q$ produziert, entstand aus der Synchronisation einer Transition von N mit einem Übergang aus $A_{N, \mu}$. Daher ist jedes $q \in Q$ in $\text{REP}_{\mu}^{\pi}(N, G)$ inhärent 1-beschränkt. Folglich können wir

Perspektive	Abschnitt	Konstruktion/Einschränkungen
Identität	4.1.1	direkt
Anwendungsfall	4.1.5	direkt
Terminierung (1)	4.1.2	Jede Markierung ist Zielmarkierung oder schwach terminierend, Reduktion auf Identität
Terminierung (2)	4.1.3	Menge der Zielmarkierungen ist abgeschlossen, Reduktion auf Filter

Tabelle 1 Anwendung der Ergebnisse aus Abschnitt 4.1, S. 101

schließen, dass $\text{REP}_\mu^\pi(N, G)$ unabhängig von der Wahl von f stets ein f -Wert-Repräsentant ist.

Korollar 152

Sei $f: \wp(Q) \setminus \emptyset \rightarrow \mathbb{K}$. Dann ist $\text{REP}_\mu^\pi(N, G)$ ein zu N kostenäquivalenter f -Wert-Repräsentant.

Ist μ assoziativ, können wir stets für G den kanonischen Graphen wählen und $\text{REPAssoc}_\mu^\pi(N)$ betrachten. Aus Korollar 108, S. 77 und der Endlichkeit von N können wir folgern: $\text{REPAssoc}_\mu^\pi(N)$ ist genau dann endlich, wenn der Abschluss $(\mathbb{K}[N, \pi])^\mu$ von $\mathbb{K}[N, \pi]$ in μ endlich ist.

Korollar 153

Sei μ assoziativ. Sei $f: \wp(Q) \setminus \emptyset \rightarrow \mathbb{K}$. Sei $(\mathbb{K}[N, \pi])^\mu$ endlich. Dann ist $\text{REPAssoc}_\mu^\pi(N)$ ein endlicher, zu N kostenäquivalenter f -Wert-Repräsentant.

Wir können die Wahl der Perspektive leicht auflockern, indem wir die Resultate aus Abschnitt 4.1, S. 101 anwenden. Wir fassen die Resultate in Tabelle 1 zusammen.

4.2.2 Konstruktion von Wert-Repräsentanten für die Max-Plus-Algebra

Sei N ein gewichtetes Netz über $\mathbb{N}_0 \cup \{-\infty\}$. Wir betrachten das von der *Max-Plus-Algebra* $\text{MP} = \langle \mathbb{N}_0 \cup \{-\infty\}, \max, + \rangle$ induzierte Kostenmodell μ , sowie eine Filter-Perspektive $\pi = \text{FILTER}[T']$.

Das Netz $\text{REP}_{\text{MP}}^{T'}(N)$ (siehe Abschnitt 3.5.1.4, S. 79) ist $\{\hat{p}_{\text{ctrl}}, \hat{p}_{-\infty}\}$ -kosteneindeutig (siehe Lemma 115, S. 80) und zu N kostenäquivalent (siehe Lemma 117, S. 82). Die Kosten von N sind genau dann beschränkt, wenn \hat{p} beschränkt ist

(siehe Lemma 118, S. 82). Wir können leicht beobachten, dass $\hat{p}_{-\infty}$ beschränkt ist. Daher können wir $f = \{P \mapsto \omega\}$ wählen.

Korollar 154

$\text{REP}_{\text{MP}}^{T'}(N)$ ist ein zu N kostenäquivalenter $\{P \mapsto \omega \mid P \neq \emptyset \in \wp(\{\hat{p}, \hat{p}_{-\infty}\})\}$ -Wert-Repräsentant bezüglich π und μ .

Außerdem ist $\text{REP}_{\text{MP}}^{T'}(N)$ endlich, falls N endlich ist (siehe Korollar 120, S. 83). Also können wir schließen:

Korollar 155

Es existiert ein N' , sodass

- (1) $\llbracket N \rrbracket_{\pi}^{\mu} = \llbracket N' \rrbracket_{\pi}^{\mu}$.
- (2) N' ist ein $\{P \mapsto \omega \mid P \neq \emptyset \in \wp(\{\hat{p}, \hat{p}_{-\infty}\})\}$ -Wert-Repräsentant.
- (3) N' ist endlich.

Ebenfalls wie im vorherigen Abschnitt können wir auch hier die Wahl der Perspektive leicht auflockern, indem wir die Resultate aus Abschnitt 4.1, S. 101 wie in Tabelle 1, S. 112 beschrieben anwenden.

4.3 Schranken-Repräsentanten

In diesem Abschnitt betrachten wir die Frage, ob für ein gegebenes Netz N die Kosten unter einer gegebenen Schranke k liegen. Dazu betrachten wir ein kompositionales Verbands-Kostenmodell (siehe Definition 52, S. 49).

Es gibt Netze, bei denen die Überschreitung einer Schranke mit der Markierung eines bestimmten Platzes zusammenfällt. In solch einem Netz N ist die Entscheidung der Kostenbeschränktheit ein Erreichbarkeitsproblem¹ und damit sogar entscheidbar, wenn N unbeschränkt ist – vorausgesetzt der markierbare Platz ist bekannt. Solch ein Netz nennen wir einen *Schranken-Repräsentanten*.

Schranken-
Repräsentanten

¹Das Problem, ob ein Platz markierbar ist, kann leicht in ein Erreichbarkeitsproblem überführt werden, indem eine weitere Transition eingefügt wird, die

- (1) einmalig schalten kann, und
- (2) eine Schlinge zum fraglichen Platz aufweist.

Definition 156 (Schranken-Repräsentant)

Sei N ein gewichtetes Netz über \mathbb{K} , p ein Platz von N , μ ein Verbands-Kostenmodell über \mathbb{K} , π eine Perspektive von N und $k \in \mathbb{K}$. Dann ist N p, k -Schranken-Repräsentant bezüglich μ und π , falls gilt:

- (1) N ist k -kosteneindeutig bzgl. μ und π .
- (2) Für alle Markierungen m von N gilt: $\llbracket m \rrbracket_{\mu}^{\pi} \leq_{\mu} k$ g.d.w. $m(p) = 0$.

Für jedes Netz N existiert zudem ein äquivalenter p, k -Schranken-Repräsentant N' , das heißt, ein Netz N' , sodass

- (1) k -Kostenbeschränktheit von N und N' koinzidieren, und
- (2) N' ein p, k -Schranken-Repräsentant ist.

Dies können wir zum Beispiel zeigen, indem wir in N den Platz p einfügen und die Anfangsmarkierung wie folgt wählen: Ist N k -kostenbeschränkt, dann liegt auf p initial eine Marke, ansonsten nicht. In der Analyse ist dies jedoch nicht hilfreich, da die k -Kostenbeschränktheit entschieden werden soll, also nicht bekannt ist.

Wir betrachten daher im Folgenden das Problem aus N einen äquivalenten Schranken-Repräsentanten abzuleiten, um die Kostenbeschränktheit auf Erreichbarkeit zu reduzieren. Dabei nutzen wir insbesondere aus, dass die Verletzung einer Kostenschranke bereits durch einen verletzenden Ablauf gegeben ist.

4.3.1 Konstruktion von Schranken-Repräsentanten

Sei N ein endliches, gewichtetes Netz über \mathbb{K} , $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von N und μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$. Wir suchen nach einem zu N äquivalenten p, k -Schranken-Repräsentanten.

Allgemein können wir zu N ein kostenäquivalentes, k -kosteneindeutiges Netz (siehe Definition 122, S. 84) konstruieren. Daraus konstruieren wir einen p, k -Schranken-Repräsentanten, indem wir den Platz p einfügen und markieren, falls wenigstens eine mehrdeutige Markierung oder Markierung mit Kosten $\not\leq_{\mu} k$ erreichbar ist. Die dafür zu berechnende Menge der erreichbaren Markierungen ist jedoch im Allgemeinen unendlich. Daher ist dieser Ansatz nicht praktikabel in der Analyse.

Sei k eine Schwelle. Sei G ein k -Graph von $\mathbb{K}[N, \pi]$ in μ . Dann resultiert die Konstruktion aus Abschnitt 3.5.2.2, S. 84 in einem Netz $\text{REP}_{\mu}^{\pi}[k](N, G)$, das k -kosteneindeutig (siehe Lemma 124, S. 84), kostenäquivalent zu N (siehe

Lemma 126, S. 85) Aus Definition 71, S. 57 können wir außerdem folgern, dass $\text{REP}_\mu^\pi[k](N, G)$ ein ℓ, k -Schranken-Repräsentant ist.

Ist μ assoziativ, können wir für G den kanonischen k -Graphen annehmen, und folgern: Wenn der k -Abschluss $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ endlich ist, ist $\text{REPAssoc}_\mu^\pi[k](N)$ endlich (siehe Korollar 129, S. 86).

Diese Lösung lässt sich insbesondere in dem Fall anwenden, falls μ durch die Max-Plus-Algebra induziert wird. Wir können dann gleichermaßen die Konstruktionen aus Abschnitt 4.2.2, S. 112 verwenden und zwei Transitionen t_1 und t_2 einfügen: t_1 habe eine Schlinge mit Gewicht $k + 1$ zum Platz \hat{p} und produziere eine Marke auf den Platz p ; t_2 habe eine Schlinge zu $\hat{p}_{-\infty}$ und konsumiere eine Marke von p . Das Ergebnis ist ein p, k -Schranken-Repräsentant, da t schalten kann g.d.w. die Schranke k überschritten wurde².

Im Folgenden betrachten wir Möglichkeiten, die obigen Anforderungen an die Perspektive π zu lockern, indem wir die Ergebnisse aus Abschnitt 4.1, S. 101 anwenden. Dabei können wir wie in Abschnitt 4.2, S. 109 vorgehen, Tabelle 1, S. 112 fasst die Möglichkeiten zusammen.

4.4 Diskussion

Wir diskutieren kurz die Ergebnisse dieses Kapitels.

4.4.1 Perspektiven-Repräsentanten und Filter-Perspektiven

Wir können Perspektiven-Repräsentanten als eine Rechtfertigung dafür verstehen, dass wir in der verbleibenden Arbeit vor allem Filter-Perspektive betrachten, da wir für mehrere Perspektiven erklärt haben, wie sie auf Filter-Perspektiven reduziert werden können. Filter-Perspektiven sind technisch sehr einfach zu handhaben, da hier jede Folge von Transitionen einfach durch Projektion auf eine neue Folge abgebildet wird. Außerdem spielen Filter-Perspektiven eine besondere Rolle bei der Komposition offener Systeme und der Betrachtung der Kosten einer Komponente (siehe Abschnitt 7.1, S. 137).

²Technisch stimmt das nicht ganz, da die Markierung von p verzögert eintreten kann; das heißt, jeder Ablauf w , der die Kostenschranke verletzt, kann so verlängert werden, dass p markiert wird

4.4.2 Kosteneindeutigkeit als Grundlage für Wert-Repräsentanten und Schranken-Repräsentanten

Die Begriffe des Wert-Repräsentanten und des Schranken-Repräsentanten fordern die Kosteneindeutigkeit eines Netzes. Das Ziel dahinter ist es, nicht-funktionale Aspekte (Kosten) auf funktionale Aspekte (erreichbare Markierungen) zu übertragen. Den besonderen Nutzen klären wir in Kapitel 5.

4.4.3 Wert-Repräsentanten und Beschränktheit

Wir haben uns entschieden, den Begriff des Wert-Repräsentanten mit einem Parameter f zu versehen, der jede Menge von Plätzen des gegebenen Netzen auf einen Wert abbildet. Diese Funktion f ist ausschließlich in dem Falle interessant, in dem das Netz unbeschränkt ist. Die Idee dahinter ist, dass wir Wert-Repräsentanten konstruieren können, für die wir wissen, dass sie entweder beschränkt sind, oder fehlende Beschränktheit ein Indiz für unbeschränkte Kosten sind. Konkret wissen wir beispielsweise im Falle der Max-Plus-Algebra, dass die Unbeschränktheit des Platzes \hat{p} mit der Unbeschränktheit der Kosten zusammenfällt (siehe Lemma 119, S. 83).

5 Analyse auf Basis von Repräsentanten

In diesem Kapitel stellen wir Entscheidungs- und Berechnungsverfahren vor, deren Korrektheit wir mithilfe der Resultate aus Kapitel 4, S. 101 beweisen. In Abschnitt 5.1 stellen wir unsere Lösung zur Berechnung der Wert-Semantik vor, in Abschnitt 5.2, S. 120 betrachten wir die k -Kostenbeschränktheit und in Abschnitt 5.3, S. 123 die Kostenbeschränktheit. Schließlich diskutieren wir unsere Resultate in Abschnitt 5.4, S. 126

5.1 Berechnung der Wert-Semantik mit Repräsentanten

In diesem Abschnitt betrachten wir die Berechnung der Wert-Semantik eines gegebenen Netzes mithilfe von Wert-Repräsentanten. Abbildung 24 zeigt abstrakt unsere Vorgehensweise: Zu einem gegebenen Netz N konstruieren wir zunächst einen äquivalenten f -Wert-Repräsentanten N' . Anschließend konstruieren wir den Überdeckbarkeitsgraphen G von N [KM69; Mur89; Fin93; Sch99]. Ist die Menge P der unbeschränkten Plätze von N leer, können wir die Wert-Semantik mit einer einfachen Tiefensuche in G lösen. Ansonsten ist das Ergebnis $f(P)$.

Sei in diesem Abschnitt N ein endliches, gewichtetes Netz über \mathbb{K} mit Transitionen T . Sei π eine kompositionale Perspektive von N und μ ein kompositionales, assoziatives Kostenmodell über \mathbb{K} .

Wir betrachten die Umstände, unter denen wir die Wert-Semantik von N berechnen können. Zunächst zeigen wir die Berechenbarkeit der Wert-Semantik von Wert-Repräsentanten (siehe Abschnitt 4.2, S. 109).

Lemma 157

Sei P eine Menge von Plätzen von N und $f: \wp(P) \setminus \emptyset \rightarrow \mathbb{K}$ eine Abbildung. Ist N ein f -Wert-Repräsentant bezüglich π und μ , ist die Wert-Semantik $\llbracket \pi \rrbracket_{\mu}^{\pi}$ berechenbar.

Beweis (Lemma 157)

Wir unterscheiden zwei Fälle:

- (1) Seien alle Plätze $p \in P$ in N beschränkt. Sei M die Menge der erreichbaren Markierungen von N . Sei \sim die Äquivalenzrelation

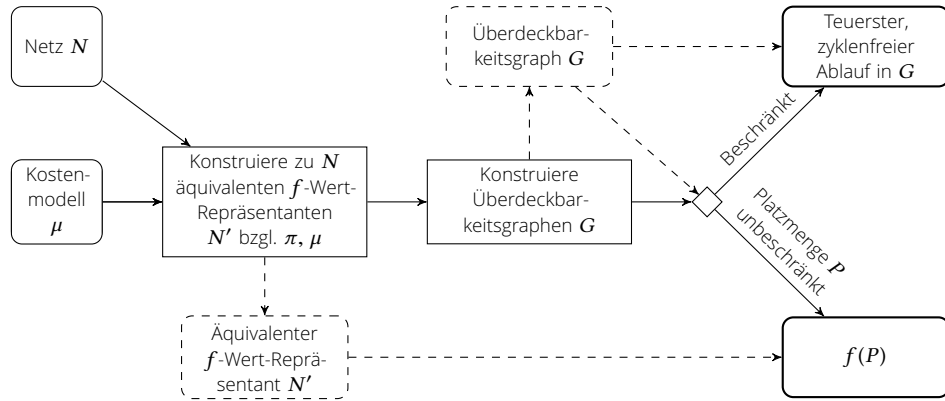


Abbildung 24 Unser Ansatz zur Berechnung der Wert-Semantik mit Wert-Repräsentanten

auf M mit $m \sim m'$ g.d.w. $m|_P = m'|_P$. Sei $[m]$ die Äquivalenzklasse von m bezüglich \sim und $[M]$ die Menge aller $[m]$ mit $m \in M$. Da N ein f -Wert-Repräsentant ist, ist N P -kosteneindeutig. Also können wir $\llbracket N \rrbracket_\mu^\pi$ aus $[M]$ berechnen. Da alle $p \in P$ beschränkt sind, ist $[M]$ eine endliche Menge. Da N ein endliches Netz ist, lässt sich $[M]$ mithilfe des Überdeckbarkeitsgraphen berechnen.

- (2) Sei wenigstens ein $p \in P$ unbeschränkt in N und P' die Menge unbeschränkten Plätze aus P in N . Dann gilt $f(P') = \llbracket N \rrbracket_\mu^\pi$. Die Menge P' lässt sich aus dem Überdeckbarkeitsgraphen von N bestimmen.

Da Beschränktheit eines Platzes p in einem Netz N entscheidbar ist, können wir zwischen den beiden Fällen unterscheiden. In beiden Fällen haben wir gezeigt, dass die $\llbracket N \rrbracket_\mu^\pi$ berechenbar ist. Also ist $\llbracket N \rrbracket_\mu^\pi$ unter den Voraussetzungen berechenbar.

Da nicht jedes Netz N ein Wert-Repräsentant ist, ist dieser Ansatz nicht universell. Wir können jedoch unter bestimmten Umständen einen endlichen Wert-Repräsentant konstruieren. Im folgenden Beweis verwenden wir die Ergebnisse aus Abschnitt 4.2.2, S. 112.

Lemma 158

Existiere ein Algorithmus, der N in einen zu N kostenäquivalenten f -Wert-Repräsentanten N' überführt. Dann ist die Wert-Semantik von N berechenbar.

Beweis (Lemma 158)

Wir berechnen zunächst N' . Aus Lemma 157, S. 117 folgt, dass die Wert-

Semantik von N' berechenbar ist. Da N' zu N kostenäquivalent ist, ist die Wert-Semantik von N ebenfalls berechenbar.

Zunächst betrachten wir eine Filter-Perspektive $\pi = \text{FILTER}[T']$ und ein assoziatives Kostenmodell über einem Verband. Ist der Abschluss von $\mathbb{K}[N, \pi]$ in μ endlich, können wir einen Wert-Repräsentanten $\text{REPAssoc}_\mu^\pi(N)$ wie folgt berechnen:

- (1) Berechne den kanonischen Graphen G von $\mathbb{K}[N, \pi]$ in μ . Dieser ist laut Korollar 70, S. 56 endlich.
- (2) Berechne $\text{REPAssoc}_\mu^\pi(L)$ aus L und G . Dies findet direkt auf der Struktur statt und das Resultat ist laut Korollar 129, S. 86 endlich.

Daher können wir zeigen:

Satz 159 (Berechenbarkeit der Wert-Semantik (Verband))

Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von N . Sei $\mathbb{K}[N, \pi] = \{\llbracket t \rrbracket_N \mid t \in T'\}$. Sei $(\mathbb{K}[N, \pi])^\mu$ endlich. Dann ist die Wert-Semantik von N bezüglich π und μ berechenbar.

Beweis (Satz 159)

Laut Satz 109, S. 77 resultiert die Konstruktion aus Abschnitt 4.2.1, S. 111 resultiert in einem endlichen, zu N kostenäquivalenten f -Wert-Repräsentanten $\text{REPAssoc}_\mu^\pi(N)$. Wir können also Lemma 158, S. 118 anwenden.

Ähnlich können wir bei dem durch die Max-Plus-Algebra konstruierten Kostenmodell verfahren. Die Konstruktion von $\text{REP}_{\text{MP}}^T(N)$ vergrößert das Netz N um endlich viele Elemente.

Satz 160 (Berechenbarkeit der Wert-Semantik (Max-Plus))

Sei π eine Filter-Perspektive von N . Sei μ durch die Max-Plus-Algebra induziert. Dann ist die Wert-Semantik von N bezüglich π und μ berechenbar.

Beweis (Satz 160)

Sei $\pi = \text{FILTER}[T]$. Laut Satz 121, S. 83 resultiert die Konstruktion aus Abschnitt 4.2.2, S. 112 in einem zu N kostenäquivalenten, endlichen $\{P \mapsto \omega \mid P \neq \emptyset \in \wp(\{\hat{p}, \hat{p}_{-\infty}\})\}$ -Wert-Repräsentanten $\text{REP}_{\text{MP}}^T(N)$. Wir können also Lemma 158, S. 118 anwenden.

Wenn wir zusätzlich die Resultate aus Abschnitt 4.1, S. 101 entsprechend Tabelle 1, S. 112 anwenden, können wir diese Aussage ausweiten.

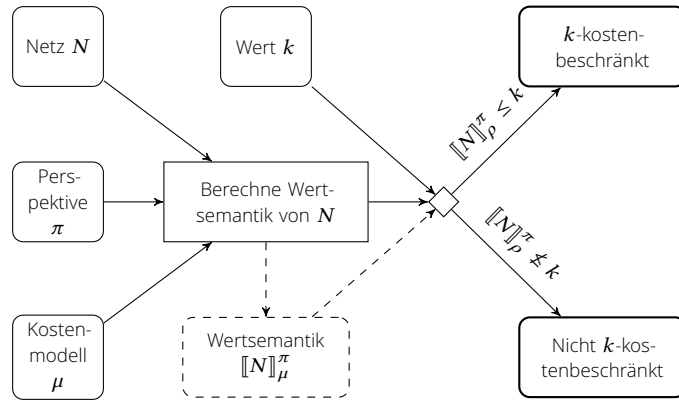


Abbildung 25 Unser Ansatz zur Entscheidung der k -Kostenbeschränktheit durch Berechnung der Wert-Semantik

Korollar 161

Sei $T' \subseteq T$ eine Menge von Transitionen von N und U ein Anwendungsfall von N . Sei

$$\pi \in \{\text{FILTER}[T'], \text{ID}, \text{AF}[N, U]\} \quad .$$

Sei $\mathbb{K}[N, \pi]$ die Menge der in N vorkommenden Kostenwerte. Sei μ durch die Max-Plus-Algebra induziert oder $(\mathbb{K}[N, \pi])^\mu$ endlich. Dann ist die Wert-Semantik von N bezüglich π und μ berechenbar.

Korollar 162

Sei $\pi = \text{TERM}[N]$. Sei $\mathbb{K}[N, \pi]$ die Menge der in N vorkommenden Kostenwerte. Sei μ durch die Max-Plus-Algebra induziert oder $(\mathbb{K}[N, \pi])^\mu$ endlich. Falls wenigstens eine der beiden folgenden Bedingungen erfüllt ist, ist die Wert-Semantik von N berechenbar:

- (1) N ist schwach terminierend.
- (2) Die Menge der Zielmarkierungen von N ist abgeschlossen unter Überdeckung und endlich erzeugt.

5.2 Entscheidung der k -Kostenbeschränktheit mit Repräsentanten

In diesem Abschnitt entscheiden wir die k -Kostenbeschränktheit mit Repräsentanten. Dabei verfolgen wir zwei Ansätze: Die Entscheidung mit Wert-Repräsentanten und die Entscheidung mit Schranken-Repräsentanten. Den Ansatz zur Entscheidung mit Wert-Repräsentanten beschreiben wir abstrakt in Abbildung 25: Wir berechnen die Wert-Semantik $\llbracket N \rrbracket_\mu^\pi$ von N und entscheiden anschließend $\llbracket N \rrbracket_\mu^\pi \leq k$. Abbildung 26 visualisiert den Ansatz zur

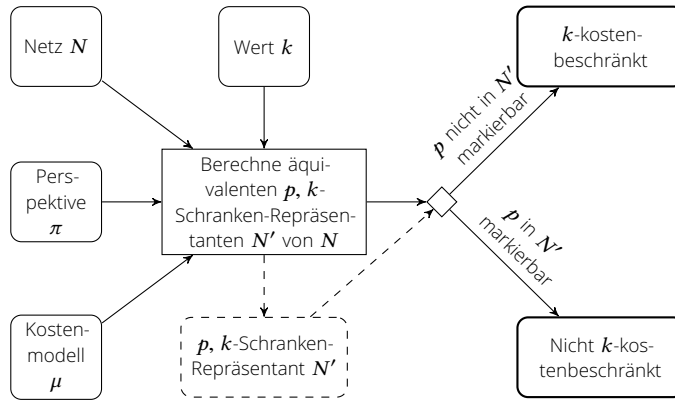


Abbildung 26 Unser Ansatz zur Entscheidung der k -Kostenbeschränktheit mit Schranken-Repräsentanten

Entscheidung mit Schranken-Repräsentanten: Hier berechnen wir im ersten Schritt einen p, k -Schranken-Repräsentanten und entscheiden anschließend, ob der Platz p markierbar ist.

Sei in diesem Abschnitt N ein endliches, gewichtetes Netz über \mathbb{K} mit Transitionen T . Sei π eine kompositionale Perspektive von N . Sei μ ein kompositionales Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ eine Schwelle. Wir nehmen dabei an, dass $\llbracket \cdot \rrbracket_N$ und μ berechenbare Funktionen sind. Wir merken an, dass dies die Entscheidbarkeit der Relation \leq_μ impliziert.

Wir betrachten die Umstände, unter denen wir die k -Kostenbeschränktheit von N entscheiden können. Als erstes beobachten wir, dass die Berechnung der Wert-Semantik ebenfalls die Entscheidung der k -Kostenbeschränktheit ermöglicht: Ist es möglich $\llbracket N \rrbracket_\mu^\pi$ zu berechnen, ist es ebenso möglich die Gültigkeit von $\llbracket N \rrbracket_\mu^\pi \leq_\mu k$ zu entscheiden. Daher können wir Korollar 161, S. 120 und Korollar 162, S. 120 aus dem vorherigen Abschnitt 5.1, S. 117 anwenden.

Korollar 163

Sei $T' \subseteq T$ eine Menge von Transitionen von N , U ein Anwendungsfall von N . Sei

$$\pi \in \{\text{FILTER}[T], \text{ID}, \text{AF}[N, U]\} \quad .$$

Sei μ durch die Max-Plus-Algebra induziert oder $(\mathbb{K}[N, \pi])^\mu$ endlich.

Dann ist die k -Kostenbeschränktheit von N bezüglich π und μ entscheidbar.

Korollar 164

Sei $\pi = \text{TERM}[N]$. Sei μ durch die Max-Plus-Algebra induziert oder die Menge $(\mathbb{K}[N, \pi])^\mu$ endlich. Falls wenigstens eine der beiden folgenden Bedingungen erfüllt ist, ist die k -Kostenbeschränktheit von N entscheidbar:

- (1) N ist schwach terminierend.
- (2) Die Menge der Zielmarkierungen von N ist abgeschlossen unter Überdeckung und endlich erzeugt.

Im weiteren Verlauf dieses Abschnittes betrachten wir die Entscheidung der k -Kostenbeschränktheit mithilfe von Schranken-Repräsentanten. Dazu zeigen wir zunächst die Entscheidbarkeit der k -Kostenbeschränktheit für Schranken-Repräsentanten (siehe Abschnitt 4.3, S. 113).

Lemma 165

Sei p ein Platz von N . Ist N ein p, k -Schranken-Repräsentant bezüglich π und μ , ist $\llbracket N \rrbracket_\mu^\pi \leq_\mu k$ entscheidbar.

Beweis (Lemma 165)

Da N ein p, k -Schranken-Repräsentant ist, gilt für alle erreichbaren Markierungen m von N : $\llbracket m \rrbracket_\mu^\pi \leq_\mu k$ g.d.w. $m(p) = 0$. Daher ist N genau dann k -kostenbeschränkt, wenn für alle erreichbaren Markierungen gilt: $m(p) = 0$, also wenn p nicht markierbar ist. Da die Markierbarkeit eines Platzes entscheidbar ist, ist auch die k -Kostenbeschränktheit entscheidbar.

Da nicht jedes Netz N ein Schranken-Repräsentant ist, ist dieser Ansatz nicht universell. Wir können jedoch unter bestimmten Umständen einen endlichen Schranken-Repräsentant konstruieren. Im folgenden Beweis verwenden wir die Ergebnisse aus Abschnitt 4.3.1, S. 114. Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive. Dann können wir den Schranken-Repräsentanten $\text{REPAssoc}_\mu^\pi[k](N)$ wie folgt berechnen:

- (1) Berechne den kanonischen k -Graphen G von $\mathbb{K}[N, \pi]$ in μ . Dieser ist laut Korollar 76, S. 59 endlich.
- (2) Berechne $\text{REPAssoc}_\mu^\pi[k](L)$ aus L und G . Dies findet direkt auf der Struktur statt und das Resultat ist laut Korollar 129, S. 86 endlich.

Satz 166 (Entscheidbarkeit der k -Kostenbeschränktheit)

Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von N . Sei der k -Abschluss $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ von $\mathbb{K}[N, \pi]$ in μ endlich. Dann ist die k -Kostenbeschränktheit von N bezüglich π und μ entscheidbar.

Beweis (Satz 166, S. 122)

Sei $\pi = \text{FILTER}[T]$. Laut Satz 130, S. 86 resultiert die Konstruktion aus Abschnitt 4.3.1, S. 114 in einem zu N kostenäquivalenten, endlichen $k, \frac{1}{2}$ -Schranken-Repräsentanten $\text{REPAssoc}_\mu^\pi[k](N)$. Aus Lemma 165, S. 122 schließen wir die Entscheidbarkeit der k -Kostenbeschränktheit des Netzes $\text{REPAssoc}_\mu^\pi[k](N)$. Da N und $\text{REPAssoc}_\mu^\pi[k](N)$ kostenäquivalent sind, ist die k -Kostenbeschränktheit von N ebenfalls entscheidbar.

Wenn wir zusätzlich die Resultate aus Abschnitt 4.1, S. 101 entsprechend Tabelle 1, S. 112 anwenden, können wir diese Aussage ausweiten.

Korollar 167

Sei $T' \subseteq T$ eine Menge von Transitionen von N und U ein Anwendungsfall von N . Sei

$$\pi \in \{\text{FILTER}[T], \text{ID}, \text{AF}[N, U]\} \quad .$$

Sei $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ endlich. Dann ist die k -Kostenbeschränktheit von N bezüglich π und μ entscheidbar.

Korollar 168

Sei $\pi = \text{TERM}[N]$. Sei $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ endlich. Falls wenigstens eine der beiden folgenden Bedingungen erfüllt ist, ist die k -Kostenbeschränktheit von N entscheidbar:

- (1) N ist schwach terminierend.
- (2) Die Menge der Zielmarkierungen von N ist abgeschlossen unter Überdeckung und endlich erzeugt.

5.3 Entscheidung der Kostenbeschränktheit mit Repräsentanten

In diesem Abschnitt untersuchen wir zwei Ansätze für die Entscheidung der Kostenbeschränktheit. Abbildung 27 zeigt den ersten Ansatz: Wir berechnen zunächst die Wert-Semantik $\llbracket N \rrbracket_\mu^\pi$ und entscheiden anschließend $\llbracket N \rrbracket_\mu^\pi < \top$ für das Top-Element \top . Der zweite Ansatz ist in Abbildung 28 visualisiert: Wir berechnen zunächst alle Schwellen k und überprüfen anschließend k -Kostenbeschränktheit für jede Schwelle k .

Sei in diesem Abschnitt N ein endliches, gewichtetes Netz über \mathbb{K} mit Transitionen T . Sei π eine kompositionale Perspektive von N . Sei μ ein kompositionales Verbands-Kostenmodell über \mathbb{K} . Sei \top das Supremum von \mathbb{K} . Sei $k \in \mathbb{K}$ eine Schwelle. Wir nehmen dabei an, dass $\llbracket \cdot \rrbracket_N$ und μ berechenbare Funktionen sind. Wir merken an, dass dies die Entscheidbarkeit der Relation \leq_μ impliziert.

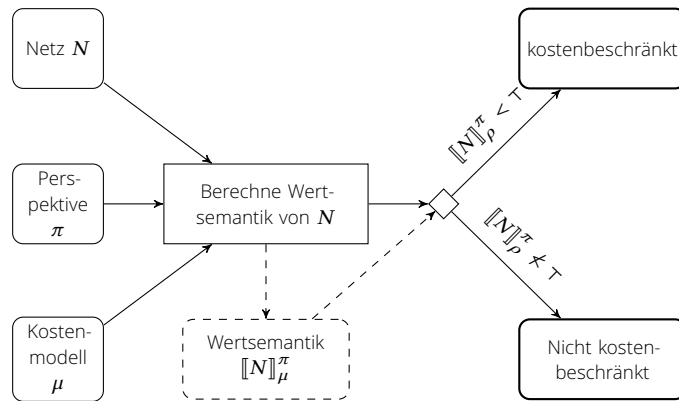


Abbildung 27 Unser Ansatz zur Entscheidung der Kostenbeschränktheit durch Berechnung der Wert-Semantik

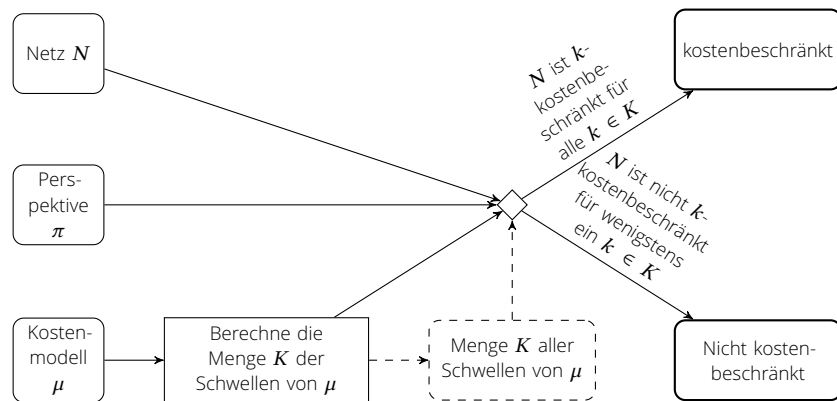


Abbildung 28 Unser Ansatz zur Entscheidung durch Überprüfung aller Schwellen

Wir betrachten die Umstände, unter denen wir die Kostenbeschränktheit von N entscheiden können. Dieses Problem ähnelt dem der Entscheidung der k -Kostenbeschränktheit für eine Schwelle k . Die Kostenbeschränktheit fordert jedoch im Gegensatz zur k -Kostenbeschränktheit für ein gegebenes k nur die Existenz eines $k \neq \top$, sodass $\llbracket N \rrbracket_{\mu}^{\pi} \leq_{\mu} k$.

Wie schon bei der k -Kostenbeschränktheit reicht die Berechnung der Wert-Semantik aus, um die Kostenbeschränktheit zu entscheiden. Daher können wir auch hier die Ergebnisse aus Abschnitt 5.1, S. 117 anwenden und zeigen:

Korollar 169

Sei $T' \subseteq T$ eine Menge von Transitionen von N und U ein Anwendungsfall von N . Sei

$$\pi \in \{\text{FILTER}[T], \text{ID}, \text{AF}[N, U]\} \quad .$$

Sei μ durch die Max-Plus-Algebra induziert oder $(\mathbb{K}[N, \pi])^\mu$ endlich.

Dann ist die Beschränktheit von N bezüglich π und μ entscheidbar.

Korollar 170

Sei $\pi = \text{TERM}[N]$. Sei μ durch die Max-Plus-Algebra induziert oder die Menge $(\mathbb{K}[N, \pi])^\mu$ endlich. Falls wenigstens eine der beiden folgenden Bedingungen erfüllt ist, ist die Beschränktheit von N entscheidbar:

- (1) N ist schwach terminierend.
- (2) Die Menge der Zielmarkierungen von N ist abgeschlossen unter Überdeckung und endlich erzeugt.

Weiterhin impliziert k -Kostenbeschränktheit für $k \neq \top$ die Kostenbeschränktheit. Enthält \mathbb{K} nur endlich viele, maximale Schwellen k , können wir durch endlich viele Entscheidungen der k -Kostenbeschränktheit auch die Kostenbeschränktheit entscheiden. Sei dazu K die Menge der Schwellen $k \in \mathbb{K}$ für die gilt:

- (1) $k \neq \top$.
- (2) Für alle $x \neq \top \in \mathbb{K}$ gilt: $k \not\leq_\mu x$.

Korollar 171

Sei K endlich. Existiere für alle Teilmengen $\mathbb{K}' \subseteq \mathbb{K}$ mit Supremum $k' \neq \top$ ein $k \in K$ mit $k' \leq_\mu k$. Sei $T' \subseteq T$ eine Menge von Transitionen von N und U ein Anwendungsfall von N . Sei

$$\pi \in \{\text{FILTER}[T], \text{ID}, \text{AF}[N, U]\} \quad .$$

Sei $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ endlich. Dann ist die k -Kostenbeschränktheit von N bezüglich π und μ entscheidbar.

Korollar 172

Sei K endlich. Existiere für alle Teilmengen $\mathbb{K}' \subseteq \mathbb{K}$ mit Supremum $k' \neq \top$ ein $k \in K$ mit $k' \leq_\mu k$. Sei $\pi = \text{TERM}[N]$. Sei $(\mathbb{K}[N, \pi])_{\leq k}^\mu$ endlich. Falls wenigstens eine der beiden folgenden Bedingungen erfüllt ist, ist die k -Kostenbeschränktheit von N entscheidbar:

- (1) N ist schwach terminierend.
- (2) Die Menge der Zielmarkierungen von N ist abgeschlossen unter Überdeckung und endlich erzeugt.

5.4 Diskussion

Wir grenzen im Folgenden die Ergebnisse dieses Kapitels gegen Methoden aus der Literatur ab.

5.4.1 Berechnung der Wert-Semantik

Die Berechnung der Wert-Semantik generalisiert das klassische *Kürzeste-Wege-Problem* und entspricht grob der gewichteten Variante dieses Problems aus dem Forschungsgebiet der gewichteten Automaten [Moh09]. Der entscheidendste Unterschied zum klassischen Problem besteht darin, dass beliebige Gewichte an den Transitionen erlaubt sind. Diese sind dann, anders als im klassischen Problem über natürlichen Zahlen, zu verrechnen. In [Moh09] wird dazu die Addition aus einem Semiring angewendet. Im Falle unendlich vieler Wege zwischen zwei Zuständen ist Vollständigkeit des Semirings erforderlich. Wir fordern nicht, dass das Kostenmodell durch einen Semiring erzeugt wird, fordern insbesondere auch keine Distributivität. Stattdessen betrachten wir als Addition stets die Bildung des Supremums bezüglich eines vollständigen Verbands. Daher sind die Forderungen unseres Ansatzes und des Ansatzes in [Moh09] orthogonal. Unser Ansatz betrachtet bewusst auch die Berechnung der Wert-Semantik für unbeschränkte Netze und nutzt dabei aus, dass die Beschränktheit von Petrinetzen entscheidbar ist. Die Forderung der Kosteneindeutigkeit erlaubt die Betrachtung des stets endlichen Überdeckbarkeitsgraphen anstatt die des im Allgemeinen unendlichen Erreichbarkeitsgraphen.

Weitere mögliche Herangehensweise wären statische Analyse [NNH99; GS10], abstrakte Interpretation [Cou96] oder Techniken aus Spieltheorie [Bri+11].

5.4.2 Entscheidung der Kostenbeschränktheit und der k -Kostenbeschränktheit

Als erste Lösung schlagen wir die Berechnung der Wert-Semantik vor und einen anschließenden Vergleich mit einem gegebenen k oder dem Top-Element \top des Verbandes vor. Bei diesem Ansatz nutzen wir nicht aus, dass wir nicht an den tatsächlichen Kosten sondern nur dem Vergleich mit einem k oder \top interessiert sind. Als zweite Lösung schlagen wir eine Reduktion auf ein Erreichbarkeitsproblem vor. Da Erreichbarkeit in Petrinetzen entscheidbar ist, können wir auch hier unbeschränkte Netze analysieren. Für die Lösung solcher Erreichbarkeitsprobleme existieren zahlreiche Techniken aus der Analyse von Petrinetzmodellen [Wol14].

6 Exkurs: Analyse von Multihemiring-Automaten

Dieses Kapitel fasst die Ergebnisse einer Zusammenarbeit mit Simon Heiden und Marvin Triebel zusammen. Für detaillierte Definitionen und Beweise verweisen wir auf [HST14].

In diesem Abschnitt zeigen wir, wie Multi-Hemiringe in Hemiringe und Semiringe eingebettet werden können, um Resultate aus der Theorie [DK09; Sak09] und der Algorithmik [Moh09] für Semiringe auf Multi-Hemiringe zu übertragen.

Gewichtete Sprachen (auch: *quantitative Sprachen*, *formale Potenzreihen*) [Sch61; Eil74; Sal+78; KLo7; CDH08; DKV09] generalisieren formale Sprachen, indem jedem Wort über einem Alphabet ein Wert aus einer Domäne anstatt eines Wahrheitswerts zugeordnet wird: Eine Abbildung $\ell: \Sigma^+ \rightarrow \mathbb{K}$ von der Menge der Wörter¹ über einem Alphabet Σ in ein Monoid $M = \langle \mathbb{K}, +, 0 \rangle$ ist eine *gewichtete Sprache* über Σ und M . Wir können ℓ immer eine kanonische Sprache L über Σ zuordnen mit $L = \{w \mid \ell(w) \neq 0\}$. Diese Sprache L nennen wir den *Support* von ℓ . Analog zur Vereinigung formaler Sprachen können wir die Addition $+$ aus M auf gewichtete Sprachen über M erweitern, indem wir $\ell_1 + \ell_2$ durch $w \mapsto \ell_1(w) + \ell_2(w)$ für alle $w \in \Sigma^+$ definieren. Ist M ein Multi-Hemiring $\langle \mathbb{K}, +, \circ, 0 \rangle$ mit $\circ = (\cdot_{m,n})_{m,n \geq 1}$, können wir außerdem das *Cauchy-Produkt* $\ell_1 \circ \ell_2$ von zwei gewichteten Sprachen ℓ_1, ℓ_2 wie folgt definieren:

$$w \mapsto \sum_{uv=w} \ell_1(u) \cdot_{|u|,|v|} \ell_2(v) \quad . \quad (72)$$

Droste und Kuich [DK13] haben gezeigt, dass die Menge aller gewichteten Sprachen über Σ und M mit der Addition und dem Cauchy-Produkt einen Hemiring bildet, wobei die Sprache mit leerem Support das neutrale Element bildet. Wie üblich steht ℓ^n ($n \in \mathbb{N}_1$) für $\ell^1 = \ell$ und $\ell^{n+1} = \ell \circ \ell^n$. Wir können zeigen, dass $\ell^n(w) = 0$ für alle $n > |w|$ gilt. Da 0 das neutrale Element bezüglich der Addition ist, können wir daher auch den *Kleene-Plus-Operator* definieren:

$$\ell^+ = \sum_{n \in \mathbb{N}_1} \ell^n \quad . \quad (73)$$

Wir nennen Addition, Multiplikation und Kleene-Plus auch die *rationalen Operationen*. Auf Basis dieser Operationen können wir den Begriff der *Rationalität* [BR88] einer gewichteten Sprache definieren: Sei $X \subseteq \mathbb{K}$. Dann definieren

¹In diesem Abschnitt beschränken wir uns auf nichtleere Wörter.

wir die Menge der X -rationalen gewichteten Sprachen über Σ und M induktiv wie folgt:

- (1) Sei ℓ eine gewichtete Sprache über Σ und M , sodass für den Support L von ℓ gilt: $L = \{a\}$ mit $a \in \Sigma$ und $\ell(a) \in X$. Dann ist ℓ eine X -rationale gewichtete Sprache über Σ und M .
- (2) Seien ℓ_1 und ℓ_2 rationale gewichtete Sprachen über Σ und M . Dann sind $\ell_1 + \ell_2$, $\ell_1 \circ \ell_2$ und ℓ^+ rationale gewichtete Sprachen über Σ und M .

Gilt $X = \mathbb{K}$, nennen wir eine X -rationale gewichtete Sprache ℓ über Σ und M auch einfach *rational*.

Das bekannte Theorem von Kleene [Kle56] besagt, dass die regulären Sprachen genau die Sprachen sind, die von endlichen Automaten erkannt werden. Schützenberger [Sch61] zeigte, dass die rationalen gewichteten Sprachen über M durch die *gewichteten Automaten* über M charakterisiert werden, falls M ein Semiring ist. Dieses Resultat wurde von Droste und Kuich [DK13] schließlich auf die Fälle, in denen M ein Multi-Hemiring ist, verallgemeinert. Ein gewichteter Automat A über Σ und einem Multi-Hemiring M ist dabei ein endlicher Automat, in dem jede Kante ein Gewicht aus \mathbb{K} aufweist. Mit Hilfe der Produkte von M können wir jedem Pfad $p = p_1 \dots p_n$ einen Wert x_p aus \mathbb{K} zuordnen: Sei x_i das Gewicht von p_i . Dann definieren wir $x_p = \circ^+(x_1 \dots x_n)$ (siehe Notation 8, S. 23). Das *Verhalten* von A ist die gewichtete Sprache mit

$$w \mapsto \sum_{p \text{ ist ein } w\text{-akzeptierender Pfad in } A} x_p . \quad (74)$$

Der generalisierte Satz von Schützenberger besagt:

Behauptung 173 (Droste und Kuich [DK13])

Eine gewichtete Sprache ℓ über Σ und M ist rational g.d.w. ein gewichteter Automat mit Verhalten ℓ existiert.

Das erste Ziel unserer Einbettung besteht in einem alternativen Beweis für diese Charakterisierung der rationalen gewichteten Sprachen über einem Multi-Hemiring M durch die gewichteten Automaten über M . Das zweite Ziel besteht darin, Algorithmen für gewichtete Automaten über Semiringen [Moh09] auf gewichtete Automaten über Multi-Hemiringen zu übertragen.

Wir definieren die Einbettung eines Multi-Hemirings M in einen Semiring S in zwei Stufen: Zunächst betten wir M in einen Hemiring H ein. Anschließend wenden wir die von Golan [Gol99] beschriebene Dorroh-Erweiterung [Dor32] auf H an, die in einem Semiring S resultiert. Die Herausforderung bei der Einbettung von M in H besteht darin, die unendliche Familie \circ der Produkt-Operationen zu einem einzigen Produkt \cdot zusammenzufassen. Die Einbettung

von H in S besteht im Wesentlichen darin, ein neutrales Element für dieses Produkt \cdot einzuführen.

Wir betten M in einen Hemiring H ein, indem wir als neue Trägermenge die Menge $\mathbb{K}_1^{\mathbb{N}}$ aller Funktionen $f: \mathbb{N}_1 \rightarrow \mathbb{K}$ betrachten². Die Operationen $+$ und \circ auf $\mathbb{K}_1^{\mathbb{N}}$ definieren wir analog zu Addition und Cauchy-Produkt auf gewichteten Sprachen. Genau wie die gewichteten Sprachen über Σ und M bildet $\mathbb{K}_1^{\mathbb{N}}$ mit den Operationen $+$ und \circ einen Hemiring mit der konstanten Abbildung $\mapsto \mathbb{0}$ mit $n \mapsto \mathbb{0}$ für alle $n \in \mathbb{N}_1$ als neutralem Element:

Behauptung 174 ([HST14])

$H = \langle \mathbb{K}_1^{\mathbb{N}}, +, \circ, \mapsto \mathbb{0} \rangle$ ist ein Hemiring.

Wir betten H in einem Semiring S wie folgt ein: Als Träger betrachten wir die Menge $\mathbb{N}_1 \times \mathbb{K}_1^{\mathbb{N}}$. Wir definieren die Addition punktweise:

$$\langle x, f \rangle + \langle y, g \rangle = \langle x + y, f + g \rangle . \quad (75)$$

Die Multiplikation definieren wir, indem wir $\langle x, f \rangle$ als die abstrakte Summe aus x und f auffassen und das Distributivgesetz anwenden:

$$\langle x, f \rangle \circ \langle y, g \rangle = \langle x \circ y, yf + xg + f \circ g \rangle , \quad (76)$$

wobei $yf = \sum_{i=1}^y f$ und $xg = \sum_{i=1}^x g$. Dann ist $\langle 0, \mapsto \mathbb{0} \rangle$ ein neutrales Element der Addition und $\langle 1, \mapsto \mathbb{0} \rangle$ ein neutrales Element des Produkts. Außerdem gilt:

Behauptung 175 ([HST14])

$S = \langle \mathbb{N}_1 \times \mathbb{K}_1^{\mathbb{N}}, +, \cdot, \langle 0, \mapsto \mathbb{0} \rangle, \langle 1, \mapsto \mathbb{0} \rangle \rangle$ ist ein Semiring.

Wir können zu jeder gewichteten Sprache ℓ über M und Σ stets die kanonische gewichtete Sprache ℓ_H über Σ und H bilden mit:

$$\ell_H(w)(x) = \begin{cases} \ell(w), & \text{falls } x = |w|, \\ \mathbb{0}, & \text{sonst} , \end{cases} \quad (77)$$

Außerdem können wir zu jeder Sprache ℓ über H die kanonische gewichtete Sprache ℓ_S über Σ und S bilden mit

$$\ell_S(w) = \langle 0, \ell_H(w) \rangle . \quad (78)$$

Anders herum können wir zu einer gewichteten Sprache ℓ über S eine kanonische gewichtete Sprache $\ell_{S^{-1}}$ über H bilden, indem wir jedes Wort w mit

²Da Σ^+ für endliche Alphabete Σ abzählbar ist, entspricht f also konzeptuell der Menge der gewichteten Sprachen über Σ und M , wobei die Wahl des Alphabets unerheblich ist.

$\ell(w) = \langle x, f \rangle$ auf f abbilden. Dann gilt für eine gewichtete Sprache ℓ über Σ und H offensichtlich $(\ell_S)_{S^{-1}} = \ell$. Die Transformation einer Sprache ℓ über H in eine Sprache über M ist jedoch nicht gleichermaßen trivial, da die folgende Summe nicht für alle $f \in \mathbb{K}_1^{\mathbb{N}}$ definiert ist:

$$\sum_{n \in \mathbb{N}_1} f(n) . \quad (79)$$

In [HST14] geben wir die Bedingungen an, unter denen die Summe definiert ist. Betrachten wir eine gewichtete Sprache ℓ über Σ und H und ist die Summe (79) für alle $f = \ell(w)$ und $w \in \Sigma^+$ definiert, können wir $\ell_{H^{-1}}$ durch

$$w \mapsto \sum_{n \in \mathbb{N}_1} \ell(w)(n) \quad (80)$$

definieren. Außerdem können wir beobachten, dass für eine gewichtete Sprache ℓ über Σ und M die Sprachen $(\ell_H)_{H^{-1}}$ und $((\ell_H)_S)_{S^{-1}H^{-1}}$ stets definiert sind. Außerdem gilt $(\ell_H)_{H^{-1}} = ((\ell_H)_S)_{S^{-1}H^{-1}} = \ell$, da $\ell_H(w)(n) = 0$ für alle $n \neq |w|$.

In [HST14] zeigen wir, dass unsere Einbettungen Kommutativität und Idempotenz von M erhalten, sowie Vollständigkeit von M partiell erhalten wird.

Wir können den folgenden Zusammenhang zwischen Rationalität von ℓ , ℓ_H und ℓ_S zeigen:

Behauptung 176 ([HST14])

Sei $X \subseteq \mathbb{K}$. Sei X_H die Menge aller $f \in \mathbb{K}_1^{\mathbb{N}}$ mit $f(n) = 0$ für alle $n > 1$ und $f(1) \in X$. Sei X_S die Menge der $\langle 0, f \rangle$ mit $f \in X_H$. Sei ℓ eine gewichtete Sprache über Σ und M und ℓ' eine Sprache über Σ und H . Dann gilt:

- (1) ℓ ist X -rational g.d.w. ℓ_H X_H -rational ist.
- (2) ℓ' ist X_H -rational g.d.w. ℓ'_S X_S -rational ist.
- (3) ℓ ist X -rational g.d.w. $(\ell_H)_S$ X_S -rational ist.

Analog zu der Definition der gewichteten Sprachen ℓ_H und ℓ_S können wir zu einem Automaten A über Σ und M den kanonischen Automaten A_H über H bilden, indem wir die Gewichte wie folgt verändern: Sei $x \in \mathbb{K}$ das Gewicht von Kante e in A , dann weisen wir e in A_H stattdessen das Gewicht f zu mit

$$f(x) = \begin{cases} x & \text{falls } x = 1 \\ 0 & \text{sonst} \end{cases} . \quad (81)$$

Einen Automaten A über H können wir kanonisch in einen Automaten A_S über S überführen, indem wir jedes Gewicht x durch $\langle 0, x \rangle$ ersetzen. Dann können wir zeigen:

Behauptung 177 ([HST14])

Sei A ein Automat über Σ und M . Sei A' ein Automat über Σ und H . Sei ℓ das Verhalten von A , ℓ' das Verhalten von A' . Dann gilt:

- (1) ℓ_H ist das Verhalten des Automaten A_H .
- (2) ℓ'_S ist das Verhalten des Automaten A'_S .
- (3) $(\ell_H)_S$ ist das Verhalten des Automaten $(A_H)_S$.

Behauptung 176, S. 132 und Behauptung 177 charakterisieren den Zusammenhang zwischen gewichteten Sprachen und Automaten über M , H und S . In [HST14] folgern wir aus dieser Charakterisierung die Behauptung 173, S. 130 und studieren, welche Klassen von Algorithmen für Semiring-Automaten für Multi-Hemiring-Automaten anwendbar sind. Für die Übernahme der Algorithmen beschreiben wir dabei die folgende Idee: Wir wenden einen Algorithmus für Semiring-Automaten auf einen Automaten A über Σ und M an, indem wir den Algorithmus auf $(A_H)_S$ anwenden und anschließend das Ergebnis zurück in die Welt von M übersetzen. Dabei unterscheiden wir Entscheidungs-, Berechnungs- und Transformationsalgorithmen. Allgemein lässt sich sagen, dass sich die Algorithmen nicht ohne Einschränkungen übertragen lassen, jedoch Algorithmen existieren, für die es möglich ist: Beispiele sind die Algorithmen für *Shortest Distance*, *Reversal*, *Weighted Composition*, *Sum* und *Product* aus [Moh09].

Teil III

Analyse und Synthese offener Systeme

In diesem Teil betrachten wir die in Abschnitt 1.2, S. 5 vorgestellten Problemfragen. Dazu führen wir zunächst formal die Begriffe der *Komposition*, des *Partners* und des *optimalen Partners* in Kapitel 7, S. 137 ein. Dies ermöglicht die Formalisierung der Probleme der Bedienbarkeit, der Partnersynthese und der Repräsentation in Kapitel 8, S. 165. Wir führen zwei Konzepte zur Lösung dieser Probleme ein: *Repräsentanten* und *Kostendiskriminanten* offener Systeme. In Kapitel 9, S. 167 definieren wir Repräsentanten offener Systeme und geben Konstruktionsvorschriften äquivalenter Repräsentanten für gegebene Systeme an. Diese Resultate wir in Kapitel 10, S. 181 an, um eine erste Teilmenge der Probleme zu lösen. In Kapitel 11, S. 189 führen wir *Kostendiskriminanten* ein und untersuchen ihre Existenz für bestimmte Teilklassen offener Systeme. Anschließend wenden wir Kostendiskriminanten in Kapitel 12, S. 195 an, um Lösungen für die Probleme anzugeben, die wir nicht bereits allein mit Repräsentanten in Kapitel 10, S. 181 lösen.

7 Optimale Partner offener Systeme

In diesem Kapitel definieren wir die *Komposition* (siehe Abschnitt 7.1), und die Begriffe des *Partners* (siehe Abschnitt 7.2, S. 140) sowie des *optimalen Partners* (siehe Abschnitt 7.3, S. 153). Schließlich diskutieren wir die Wahl unserer Begriffe und setzen sie in Relation zu existierenden Begriffen (siehe Abschnitt 7.4, S. 161).

7.1 Komposition

Wir definieren die *Komposition* $L \oplus R$ zweier offener Systeme L und R entlang ihrer jeweiligen *Schnittstellen*. Dadurch realisieren wir asynchronen, bilateralen Nachrichtenaustausch über unbeschränkte Puffer, wobei ein Schnittstellenplatz einen Puffer modelliert.

Mit dem Ziel die Definition technisch einfach zu halten, nehmen wir an, dass zu komponierende offene Systeme L und R *disjunkte innere Prozesse* aufweisen, das heißt:

- (1) Jeder gemeinsame Platz von L und R ist ein Schnittstellenplatz in L und in R .
- (2) Die Transitionsmengen von L und R sind disjunkt.

Diese Voraussetzung lässt sich für beliebige L und R leicht erfüllen, indem R durch eine isomorphe Kopie ersetzt wird. Der Einfachheit halber nehmen wir diese Voraussetzung als gegeben an, wenn wir $L \oplus R$ schreiben.

Wir definieren die Komposition vorwiegend durch komponentenweise Vereinigung der Netze. Dabei bleiben Beschriftungen und Kosten von Transitionen erhalten. Die Menge der Rahmen- und Zielmarkierungen wird dabei so gebildet, dass eine Rahmenmarkierung (Zielmarkierung) des Kompositum stets auf eine Rahmenmarkierung (Zielmarkierung) jeder der Komponenten zurückgeführt werden kann. Also ist das Kompositum in einer Rahmenmarkierung (Zielmarkierung) g.d.w. beide Netze in einer Rahmenmarkierung (Zielmarkierung) sind.

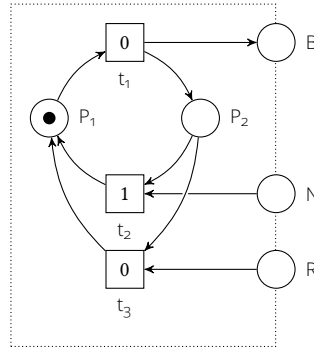


Abbildung 29 (Wiederholt aus Abbildung 12, S. 41) Ein zyklisches offenes System Net_{12} , das in jeder Runde eine neue Bestellung absendet und anschließend eine Versandoption als Antwort erwartet. Die Zielmarkierung ist $\{P_1\}$. Die Rahmenmarkierungen sind alle Markierungen m , in denen $m(p) \leq 1$ für alle $p \in \{N, B, R\}$ gilt.

Definition 178 (Komposition)

Seien L und R Netze. Die *Komposition* $L \oplus R$ von L und R ist das kleinste Netz mit folgenden Eigenschaften für $N \in \{L, R\}$:

- (1) Jeder Platz von N ist ein Platz von $L \oplus R$.
- (2) Jede Transition von N ist eine Transition von $L \oplus R$. Kosten und Beschriftungen bleiben dabei erhalten.
- (3) Die Flussfunktion von $L \oplus R$ ist die Vereinigung der Flussfunktionen von L und R .
- (4) Die Schnittstelle von $L \oplus R$ ist der Durchschnitt der jeweiligen Schnittstellen von L und R .
- (5) Die Anfangsmarkierung von $L \oplus R$ ist die Summe der Anfangsmarkierungen von L und R .
- (6) Sei P die Menge der gemeinsamen Plätze von L und R . Sei m eine Rahmenmarkierung (Zielmarkierung) von L und m' eine Rahmenmarkierung (Zielmarkierung) von R , sodass $m|_P = m'|_P$. Die Markierung $(m + m') - m|_P$ ist eine Rahmenmarkierung (Zielmarkierung) von $L \oplus R$.

Die Erhaltung der Kosten und Beschriftungen in 2. und die Vereinigung in 3. sind wohldefiniert, da per Annahme die Mengen der Transitionen von L und R disjunkt sind.

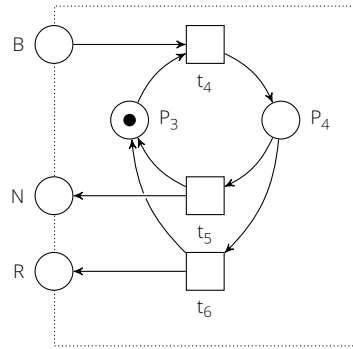


Abbildung 30 Ein zyklisches offenes System Net_{30} , das in jeder Runde eine neue Bestellung erwartet und anschließend eine Versandopation als Antwort zurücksendet. Die Zielmarkierung ist $\{P_3\}$.

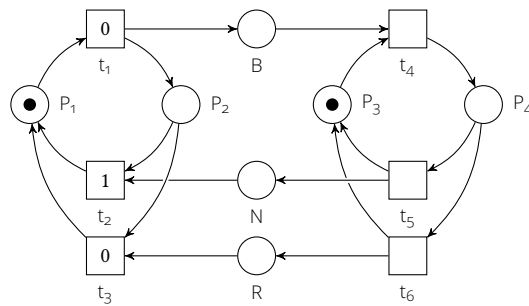


Abbildung 31 (Wiederholt aus Abbildung 13, S. 44) Die Komposition $\text{Net}_{13} = \text{Net}_{12} \oplus \text{Net}_{30}$ der offenen Systeme aus Abbildung 12, S. 41 und Abbildung 30.

Beispiel 179

Abbildung 31 zeigt die Komposition der Netze aus Abbildung 29, S. 138 und Abbildung 30. Das resultierende Netz ist geschlossen.

In Abschnitt 2.7.5, S. 32 haben wir zu jedem Netz L seine universelle Umgebung $\text{ENV}(L)$ definiert. Wir können die folgenden Eigenschaften bezüglich Komposition und Simulation (siehe Abschnitt 2.7.6, S. 33) beobachten:

Behauptung 180 ([Dzi+14])

Seien L , R und R' Netze.

- (1) $L \oplus \text{ENV}(L)$ ist geschlossen.
- (2) $L \oplus \text{ENV}(L)$ simuliert $L \oplus R$.
- (3) Gelte $R \oplus \text{ENV}(R)$ simuliert $R' \oplus \text{ENV}(R')$. Dann gilt: $L \oplus R$ simuliert $L \oplus R'$.

Perspektiven für Komposita. Wir betrachten häufig Komposita $S = L \oplus R$, wobei L fest gewählt und R quantifiziert ist. In diesem Falle ist die Definition einer Perspektive von S technisch problematisch, da sich die Menge der Transitionen und Zielmarkierungen von R von Netz zu Netz unterscheiden kann. Daher führen wir die folgende Konvention ein:

- (1) Sei T die Menge der Transitionen von L und $T' \subseteq T$, U ein Anwendungsfall von L und w ein Wort über Beschriftungen von L . Sei $\pi \in \{\text{ID}, \text{FILTER}[T], \text{AF}[L, U], \text{WFILTER}[w]\}$. Dann verstehen wir π auch als die Perspektive $\text{FILTER}[T] \triangleright \pi$ beliebiger Netze $L \oplus R$.
- (2) Sei $\pi = \text{TERM}[L]$. Dann verstehen wir die Perspektive π auch als die Perspektive $\text{TERM}[L \oplus R]$ beliebiger Netze $L \oplus R$.

Wir unterscheiden die Terminierungs-Perspektive von den anderen Perspektiven, da sich die Zielmarkierungen in der Komposition verändern.

7.2 Partner offener Systeme

In diesem Abschnitt definieren wir die Relationen *Partnerschaft*, *Bedienbarkeit* und *Ersetzbarkeit*. Anschließend betrachten wir *most-permissive Partner* in Abschnitt 7.2.1, S. 143 und die *Minimierung von Partnern* in Abschnitt 7.2.2, S. 146.

Ein offenes System R ist ein *Partner* eines offenen Systems L , falls das Kompositum $L \oplus R$ eine gegebene Anforderung erfüllt. Die Anforderung richtet sich ausdrücklich nur an das Kompositum $L \oplus R$, nicht an die einzelnen Komponenten L und R .

Definition 181 (Partner)

Seien L und R Netze. Sei ρ eine Anforderung. Dann ist R ein ρ -Partner von L , falls $L \oplus R$ die Anforderung ρ erfüllt.

Beispiel 182

Wir betrachten wir das Netz Net_{29} aus Abbildung 29, S. 138 jeweils in Komposition mit den Netzen Net_{30} aus Abbildung 30, S. 139, Net_{32} aus Abbildung 32, S. 142, Net_{33} aus Abbildung 33, S. 142 und das Schema Net_{34} aus Abbildung 34, S. 143, das die Netze Net_{34}^k beschreibt mit $k \in \mathbb{N}_0$.

Wie in Beispiel 179, S. 139 beschrieben, entspricht $\text{Net}_{29} \oplus \text{Net}_{30}$ dem Netz Net_{31} aus Abbildung 31, S. 139. Wir beobachten, dass die restlichen Komposita bereits in Abschnitt 3.4, S. 66 aufgetaucht sind:

- $\text{Net}_{29} \oplus \text{Net}_{32}$ entspricht Net_{15} aus Abbildung 15, S. 67,

- $\text{Net}_{29} \oplus \text{Net}_{33}$ entspricht Net_{17} aus Abbildung 17, S. 68 und
- Für alle $k \in \mathbb{N}_0$ gilt: $\text{Net}_{29} \oplus \text{Net}_{34}^k$ entspricht Net_{16}^k aus Abbildung 16, S. 68.

Wir betrachten zunächst $\rho = \mathbb{S}$ (siehe Notation 138, S. 89). Dabei nehmen wir an, dass die Menge der Rahmenmarkierungen jeweils die Menge aller Markierungen ist. Dann gilt für alle $k \in \mathbb{N}_0$:

$$\text{Net}_{30}, \text{Net}_{32}, \text{Net}_{33}, \text{Net}_{34}^k \in \mathbb{S}[\text{Net}_{29}] . \quad (82)$$

Verändern wir dagegen die Zielmarkierung der Netze Net_{30} , Net_{32} , Net_{33} und Net_{34}^k auf \mathcal{P}_4 , dann ist (82) nicht mehr erfüllt.

Wir betrachten nun $\rho \in \{\mathbb{B}_\mu^\pi, \mathbb{B}_\mu^\pi[k]\}$ (siehe Notation 140, S. 90). Folgen wir Beispiel 141, S. 90 beobachten wir:

- Net_{30} ist ein \mathbb{B}_μ^π -Partner von Net_{29} bezüglich $\text{FILTER}[\{t_1, t_3\}]$ jedoch *nicht* kostenbeschränkt bezüglich $\text{FILTER}[\{t_1, t_2, t_3\}]$.
- Net_{32} ist ein \mathbb{B}_μ^π -Partner von Net_{29} bezüglich $\text{FILTER}[\{t_1, t_2, t_3\}]$, jedoch *nicht* bezüglich , die jedes Vorkommen von t_1 durch ein Vorkommen von t_2 ersetzt.
- Für jedes $k \in \mathbb{N}_0$ ist Net_{34} ein $\mathbb{B}_\mu^\pi[k]$ -Partner von Net_{29} bezüglich $\text{FILTER}[\{t_1, t_2, t_3\}]$.
- Net_{33} ist ein \mathbb{B}_μ^π -Partner von Net_{29} bezüglich $\text{FILTER}[\{t_1, t_3\}]$, jedoch *nicht* bezüglich $\text{FILTER}[\{t_1, t_2, t_3\}]$.
- Sei $\Sigma = \{b, n, r\}$ und U der Anwendungsfall mit $\bullet U = \Sigma^*\{b\}$ und $U^\bullet = \{n, r\}$. Sei Net_{29}' die Variante von Net_{29} , in der die Transitionen die folgende Beschriftung aufweisen: $t_1 \mapsto b$, $t_2 \mapsto n$ und $t_3 \mapsto r$. Dann gilt: Net_{30} , Net_{33} , Net_{34} für $k > 0$ sind $\mathbb{B}_\mu^\pi[1]$ -Partner, und Net_{32} und Net_{34} für $k = 0$ sind $\mathbb{B}_\mu^\pi[0]$ -Partner von Net_{29}' .

Häufig vergleichen wir unterschiedliche Partnermengen. Daher führen wir die folgende Notation ein:

Notation 183 (Partnermengen)

Sei L ein Netz und ρ eine Anforderung. Wir schreiben $\rho[L]$ für die Menge der ρ -Partner von L .

$\rho[L]$

Wir beobachten, dass die Menge der ρ -Partner von L eine erneut eine Anforderung bildet. Wir können außerdem zeigen:

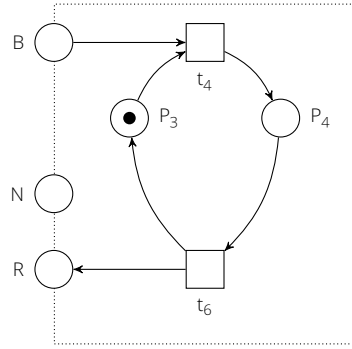


Abbildung 32 Eine Variante Net_{32} von Net_{30} aus Abbildung 30, S. 139, wobei in jeder Runde die Versandoption R als Antwort zurückgesendet wird. Die einzige Zielmarkierung ist $\{P_3\}$.

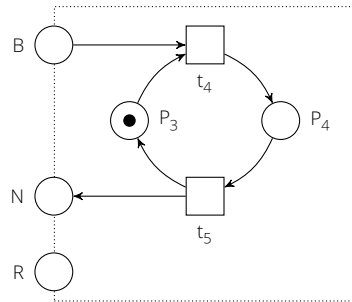


Abbildung 33 Eine Variante Net_{33} von Net_{30} aus Abbildung 30, S. 139, wobei in jeder Runde die Versandoption N als Antwort zurückgesendet wird. Die einzige Zielmarkierung ist $\{P_3\}$.

Lemma 184

Sei L ein Netz. Seien ρ, ρ' Anforderungen. Dann gilt für $* \in \{\cap, \cup, \setminus\}$:

$$(\rho * \rho')[L] = \rho[L] * \rho'[L] . \quad (83)$$

Beweis (Lemma 184)

$$(\rho * \rho')[L] = \{R \mid L \oplus R \in (\rho * \rho')\} = \{R \mid L \oplus R \in \rho\} * \{R \mid L \oplus R \in \rho'\} = \rho[L] * \rho'[L].$$

Wir definieren auf Basis des Partnerbegriffs die Begriffe *Bedienbarkeit* und *Ersetzbarkeit*: Ein offenes System ist *bedienbar*, falls es wenigstens einen Partner besitzt; ein offenes System L' *ersetzt* ein offenes System L , falls jeder Partner von L ein Partner von L' ist.

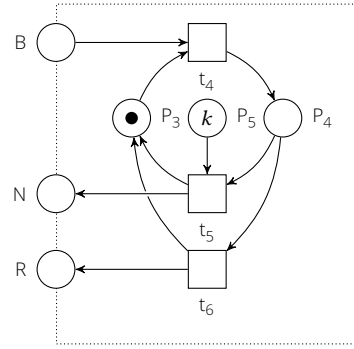


Abbildung 34 Ein Schema einer Variante Net_{34} von Net_{30} aus Abbildung 30, S. 139, wobei die Versandoption N nur k mal gesendet werden kann. Die einzige Zielmarkierung ist $\{P_3\}$. Für ein bestimmtes $k \in \mathbb{N}_0$ schreiben wir Net_{34}^k für das entsprechende Netz.

Definition 185 (Bedienbar, ersetzt)

Seien L und L' Netze. Sei ρ eine Anforderung.

- (1) Sei $\rho[L] \neq \emptyset$. L ist ρ -bedienbar.
- (2) Sei $\rho[L] \subseteq \rho[L']$. Dann sagen wir: L' ρ -ersetzt L .

Sei R ein $\rho \cap \rho'$ -Partner von L . Aus Lemma 184, S. 142 können wir leicht folgern, dass ein $\rho \cap \rho'$ -bedienbares Netz L auch ρ -bedienbar und ρ' -bedienbar ist. Wir können jedoch nicht von ρ -Bedienbarkeit und ρ' -Bedienbarkeit auf $\rho \cap \rho'$ -Bedienbarkeit schließen – die Mengen $\rho[L]$ und $\rho'[L]$ sind möglicherweise disjunkt: Dann gilt $\rho[L] \cap \rho'[L] = \rho \cap \rho'[L] = \emptyset$.

7.2.1 Most-permissive Partner

Die Menge $\rho[L]$ eines bedienbaren Netzes L bezüglich der Anforderung ρ ist für die in dieser Arbeit betrachteten Anforderungen stets unendlich. In einigen Fällen gibt es in dieser unendlichen Menge jedoch kanonische Partner. In der Literatur gibt es unterschiedliche Definitionen für solche kanonischen Partner. In dieser Arbeit nutzen wir die Existenz von *most-permissive Partnern*. Intuitiv repräsentiert ein most-permissive Partner R die Menge aller Partner, indem er in gewissem Sinne maximal ist und einen Vergleich mit Netzen R' erlaubt, um zu entscheiden, ob R' ein Partner ist [Wolog; Wol+11]. Zwei entscheidende Eigenschaften eines most-permissive Partners sind *Präzision* und *Kompaktheit*. Die Kombination aus Präzision und Kompaktheit garantiert dabei, dass R genau so viele Zustände unterscheidet, wie notwendig.

Wir führen zur Definition der Eigenschaft *most-permissive* die folgende Notation ein: Seien L und R Netze, ρ eine Anforderung und $R \in \rho[L]$. Für alle Markierungen m von $L \oplus R$ und $w \in \text{RUNS}(L \oplus R, m)$ sei

$$\text{MARKINGS}_L^R(w) = \{m' \mid v \in \text{RUNS}(L \oplus R, m'), v|_R = w|_R\} . \quad (84)$$

Das heißt, $\text{MARKINGS}_L^R(w)$ enthält genau die Markierungen von $L \oplus R$ die aus Abläufen entstehen, die aus Sicht von R jeweils aus w bestehen und damit nicht weiter unterscheidbar sind.

Beispiel 186

Als Beispiele für L und R betrachten wir jeweils die Netze Net_{29} aus Abbildung 29, S. 138 und Net_{30} aus Abbildung 30, S. 139. Für die Abläufe $v = t_1 t_4 t_5$ und $w = t_1 t_4 t_5 t_2$ gilt:

$$\text{MARKINGS}_L^R(v) = \text{MARKINGS}_L^R(w) = \{\{P_2, N, P_3\}, \{P_1, P_3\}, \{P_2, B, P_3\}\} .$$

Nun können wir die Begriffe *Präzision*, *Kompaktheit*, *Maximalität* und *most-permissive* für die Partner von L definieren. Sei P die Menge der internen Plätze von R .

- (1) Es gelte für alle Abläufe $w_i \in \text{RUNS}(L \oplus R, m_i)$ ($i = 1, 2$): $m_1|_P = m_2|_P$ impliziert $\text{MARKINGS}_L^R(w_1) = \text{MARKINGS}_L^R(w_2)$. Dann ist R *präzise*.
- (2) Es gelte für alle Abläufe $w_i \in \text{RUNS}(L \oplus R, m_i)$ ($i = 1, 2$): $m_1|_P \neq m_2|_P$ impliziert $\text{MARKINGS}_L^R(w_1) \neq \text{MARKINGS}_L^R(w_2)$. Dann ist R *kompakt*.
- (3) Es gelte für alle ρ -Partner R' : $R \oplus \text{ENV}(R)$ simuliert $R' \oplus \text{ENV}(R')$. Dann ist R *maximal*.
- (4) Gelte für alle in $R \oplus \text{ENV}(R)$ erreichbaren Markierungen m : Aktiviert m zwei Transitionen t, t' von R , dann existiert wenigstens ein Schnittstellenplatz p mit $V(p, t) \neq V(p, t')$ oder $V(t, p) \neq V(t', p)$. Dann ist R *deterministisch*.
- (5) Ist R deterministisch, präzise, kompakt und maximal, ist R *most-permissive*.

Wir merken an, dass Präzision und Kompaktheit immer für ein festes Netz L definiert sind. Maximalität ist an ein Netz L und eine Anforderung ρ gebunden. Determinismus ist unabhängig von L . Somit trägt der Begriff *most-permissive* stets ein Netz L und eine Anforderung ρ als Parameter.

Beispiel 187

Net_{30} aus Abbildung 30, S. 139 ist ein kompakter, deterministischer \mathbb{S} -Partner von Net_{29} aus Abbildung 29, S. 138. Net_{30} ist jedoch weder präzise

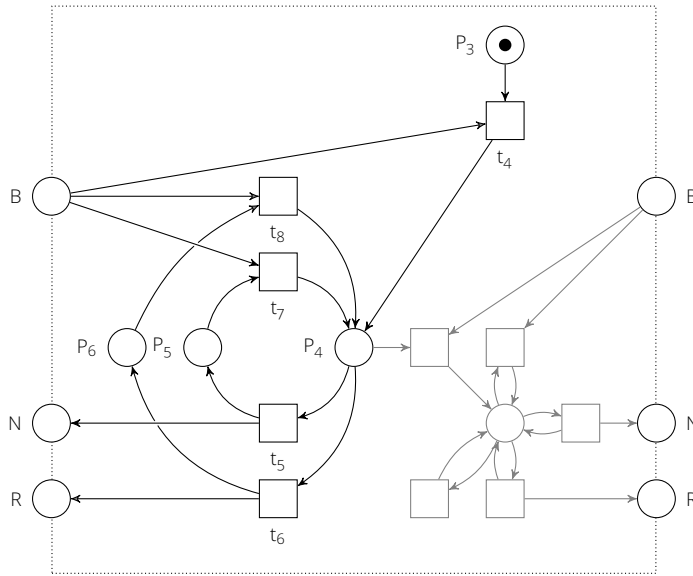


Abbildung 35 Ein most-permissive \mathbb{S} -Partner Net_{35} von Net_{29} aus Abbildung 29, S. 138. Die Schnittstellenplätze B, N und R haben wir der Übersichtlichkeit halber doppelt abgebildet. In der Komposition mit Net_{29} werden die grauen Plätze niemals markiert und die grauen Transitionen niemals aktiviert.

noch maximal: Für die Abläufe $v = t_1 t_4 t_5$ und $w = t_1 t_4 t_6$ gilt:

$$\text{MARKINGS}_L^R(v) = \{\{P_2, N, P_3\}, \{P_1, P_3\}, \{P_2, B, P_3\}\}$$

und

$$\text{MARKINGS}_L^R(w) = \{\{P_2, R, P_3\}, \{P_1, P_3\}, \{P_2, B, P_3\}\}.$$

v resultiert in $m_v = \{P_2, N, P_3\}$ und w in $m_w = \{P_2, R, P_3\}$. Also gilt zwar $m_v|_P = m_w|_P$ für $P = \{t_3, t_4\}$, jedoch $\text{MARKINGS}_L^R(v) \neq \text{MARKINGS}_L^R(w)$. Also ist Net_{30} nicht präzise. Net_{30} ist nicht maximal, da Net_{35} aus Abbildung 35 ein \mathbb{S} -Partner ist, jedoch $\text{Net}_{35} \oplus \text{Env}(\text{Net}_{35})$ nicht von $\text{Net}_{30} \oplus \text{Env}(\text{Net}_{30})$ simuliert wird.

Net_{35} aus Abbildung 35 ist jedoch ein most-permissive Partner: Im Gegensatz zu Net_{30} unterscheidet Net_{35} in seiner internen Markierung die Abläufe v und w . Die Maximalität können wir nicht so leicht am Beispiel zeigen. Intuitiv sind die grauen Plätze und Transitionen ein Mittel dafür, Verhalten auszudrücken, das mit Net_{29} nicht eintreten kann, jedoch in $R \oplus \text{Env}(R)$ für beliebige Partner R eintreten könnte.

Aus Behauptung 180, S. 139 können wir folgern, dass jede Komposition von L mit einem maximalen Partner jede Komposition mit einem beliebigem Partner

simuliert:

Korollar 188

Seien L , R und R' Netze. Seien R und R' jeweils ρ -Partner von L . Sei R maximal. Dann gilt: $L \oplus R$ simuliert $L \oplus R'$.

Aus der Literatur ist bekannt, dass unter gewissen Annahmen most-permissive Partner für schwache Terminierung existieren:

Behauptung 189

Sei L ein Netz. Sei b eine natürliche Zahl und gelte für jeden Platz p von L und jede Rahmenmarkierung m von L : $m(p) \leq b$. Dann sind die beiden folgenden Aussagen äquivalent:

- (1) $\mathbb{S}[L] \neq \emptyset$.
- (2) Es existiert ein endlicher most-permissive \mathbb{S} -Partner $R \in \mathbb{S}[L]$ von L , so dass $L \oplus R$ beschränkt ist.

Zu einem Netz können durchaus mehrere most-permissive Partner existieren. In dieser Arbeit ist die konkrete Wahl des most-permissive Partner eines Netzes unerheblich. Daher führen wir die folgende Notation ein:

Notation 190 (Most-permissive Partner)

Sei L ein Netz, sodass ein endlicher, most-permissive Partner $R \in \mathbb{S}[L]$ existiert. Dann identifizieren wir alle most-permissive Partner $R \in \mathbb{S}[L]$ und schreiben $\text{MPP}(L)$ für *den* most-permissive \mathbb{S} -Partner von L .

7.2.2 Minimierung von Partnern

Im vorherigen Abschnitt haben wir die Existenz von most-permissive Partnern betrachtet, die in gewissem Sinne maximale Elemente der Partnermenge sind. In diesem Abschnitt wenden wir uns einem gegensätzlichen Ziel zu: Gegeben ein Partner R stellen wir die Frage, wie dieser *minimiert* werden kann. Eine besondere Rolle spielt dabei die *Überdeckung von Zyklen*: Ein Netz L weist in der Komposition mit seiner universellen Umgebung eine Reihe von Zyklen auf. Ein Partner kann möglicherweise *entscheiden*, diese Zyklen zu betreten oder ist *gezwungen* sie zu betreten.

Beispiel 191

Wir betrachten die zwei Netze Net_{36} aus Abbildung 36 und Net_{37} aus Abbildung 37 jeweils in Verbindung mit ihrer universellen Umgebung.

Net_{36} zeigt zusammen mit seiner universellen Umgebung den Zyklus, in

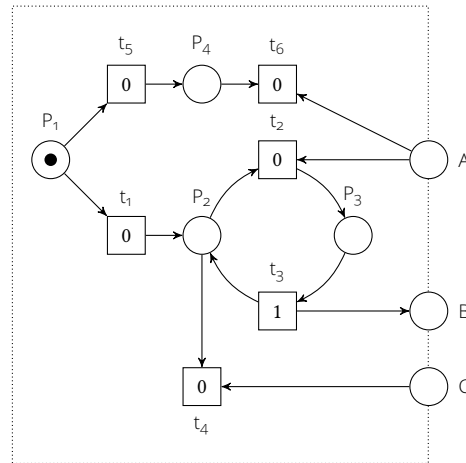


Abbildung 36 Ein zyklisches offenes System Net_{36} mit der einzigen Zielmarkierung $\{ \}$, dessen Zyklus wenigstens einmal betreten werden muss, um schwache Terminierung zu erreichen.

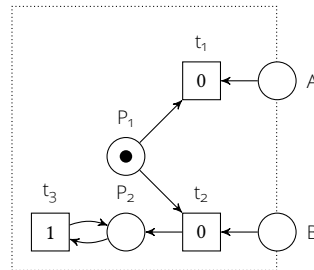


Abbildung 37 Ein zyklisches offenes System Net_{37} mit den Zielmarkierungen $\{ \}$ und $\{ P_2 \}$, bei dem der Partner entscheiden kann, ob der interne Zyklus betreten wird.

dem abwechselnd die Transition t_2 und t_3 schalten. Ein \mathbb{S} -Partner R von Net_{36} wird – intuitiv gesprochen – gezwungen, diesen Zyklus zu betreten: Net_{36} kommuniziert die Entscheidung zwischen den Transitionen t_1 und t_5 nicht. Daher hat R kein Wissen darüber, ob sich Net_{36} im oberen oder unteren Zweig befindet. Im oberen Zweig muss R die Nachricht A schicken – ansonsten ist die Zielmarkierung von Net_{36} nicht mehr erreichbar. Im unteren Zweig würde das Senden der Nachricht C ausreichen, und das Senden der Nachricht A aktiviert den Zyklus. Da R jedoch, wie gesagt, kein Wissen über den gewählten Zweig hat, muss R gezwungenermaßen Nachricht A senden. Im Falle des oberen Zweigs ist die Kommunikation dann beendet. Im Falle des unteren Zweigs bestätigt Net_{36} den Empfang der Nachricht A durch Senden der Nachricht B . Durch den Empfang dieser Nachricht findet R heraus, dass der untere Zweig gewählt wurde und

kann so die Nachricht C senden, oder den Zyklus erneut durch Senden der Nachricht A betreten.

Für die \mathbb{S} -Partner des Netzes Net_{37} sieht es dagegen anders aus: Net_{37} weist mit seiner universellen Umgebung den Zyklus auf, in dem t_3 immer wieder schaltet. Voraussetzung hierfür ist das vorherige Eintreten der Transition t_2 . Ein \mathbb{S} -Partner R hat die Wahl, t_1 oder t_2 durch das Senden der Nachricht A oder B zu aktivieren – in beiden Fällen erreicht Net_{37} eine Zielmarkierung. Daher entscheidet R über das Betreten des Zyklus.

In diesem Abschnitt zeigen wir, wie Partner *bezüglich der Überdeckung von Zyklen minimiert* werden können. Die Resultate dieses Abschnitts wenden wir in Abschnitt 11.2.1, S. 190 zur Konstruktion *kanonischer Kostendiskriminanten* an.

Sei L ein Netz. Wir definieren den Begriff des L -Zyklus und die *Überdeckung* von L -Zyklen durch Partner. Dabei betrachten wir stets die Anforderung der schwachen Terminierung. Das heißt, mit Partner meinen wir \mathbb{S} -Partner.

Wir definieren L -Zyklen und ihre Überdeckung aus der Sicht der Partner von L : Gelte $p \xrightarrow{r} p$ in $\text{MPP}(L) \oplus \text{ENV}(\text{MPP}(L))$. Dann ist $z = \langle p, r \rangle$ ein L -Zyklus. Sei R ein Partner von L und w ein Ablauf von $R \oplus \text{ENV}(R)$. Dann entspricht $w = w_1 \dots w_n$ kanonisch einem Ablauf $v = v_1 \dots v_n$ in $\text{MPP}(L) \oplus \text{ENV}(\text{MPP}(L))$, da $\text{MPP}(L)$ maximal ist, und daher $L \oplus \text{ENV}(L)$ von $\text{MPP}(L) \oplus \text{ENV}(\text{MPP}(L))$ simuliert wird. Ein Ablauf w *überdeckt* den L -Zyklus $z = \langle p, r \rangle$ k -mal, geschrieben $w[z] = k$, falls z in v genau k -mal vorkommt.

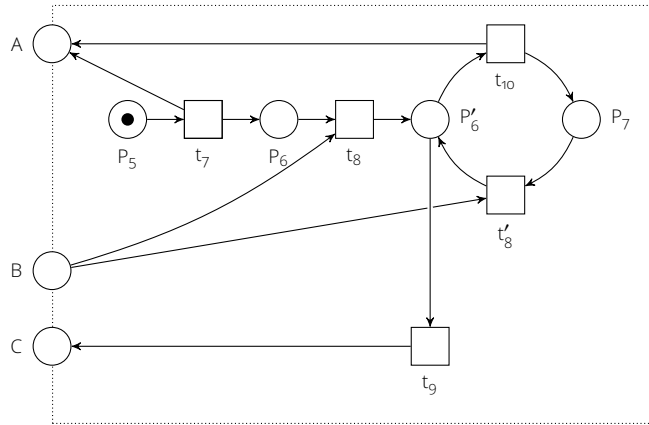


Abbildung 38 Ein most-permissive \mathbb{S} -partner Net_{38} von Net_{36} . Jede Markierung von Net_{38} ist eine Zielmarkierung. Der Übersichtlichkeit halber haben wir nur Verhalten dargestellt, das auch mit Net_{36} auftreten kann.

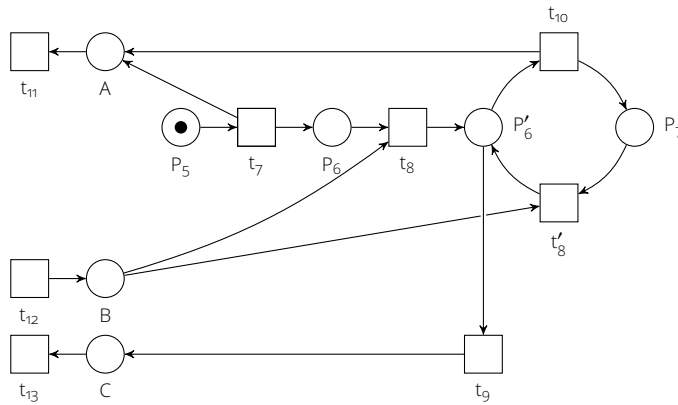


Abbildung 39 Die Komposition $\text{Net}_{39} = \text{Net}_{38} \oplus \text{Env}(\text{Net}_{38})$ des Netzes Net_{38} aus Abbildung 38, S. 148 mit seiner universellen Umgebung $\text{Env}(\text{Net}_{38})$.

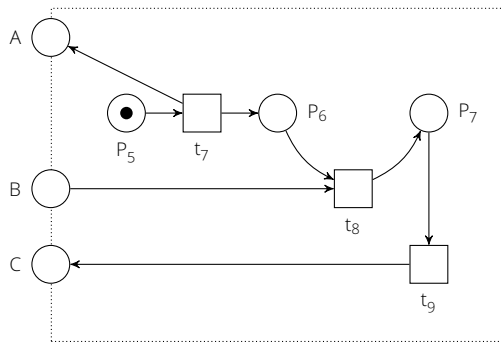


Abbildung 40 Ein \mathbb{S} -Partner Net_{40} von Net_{36} , der den Zyklus in Net_{36} nur einmal überdeckt. Jede Markierung von Net_{40} ist eine Zielmarkierung.

Beispiel 192

Wir betrachten wir das Netz Net_{36} aus Abbildung 36, S. 147 und seinen most-permissive Partner^a Net_{38} aus Abbildung 38, S. 148 sowie zwei weitere Partner Net_{40} aus Abbildung 40 und Net_{41} aus Abbildung 41.

Abbildung 39 zeigt das Netz $\text{Net}_{39} = \text{Net}_{38} \oplus \text{Env}(\text{Net}_{38})$. Darin können wir den Zyklus $z = t_{10}t_{11}t_{12}t'_8$ aus der Markierung $\{P'_6\}$ beobachten: $\{P'_6\} \xrightarrow{t_{10}} \{P_7, A\} \xrightarrow{t_{11}} \{P_7\} \xrightarrow{t_{12}} \{P_7, B\} \xrightarrow{t'_8} \{P'_6\}$. Da Net_{38} der most-permissive Partner von Net_{36} ist, ist z ein Net_{36} -Zyklus.

Das Netz Net_{40} überdeckt diesen Zyklus z nicht. Für jedes $k \in \mathbb{N}_0$ überdeckt das Netz Net_{41} den Zyklus bis zu k -mal: Sei $k = 3$. Dann über-

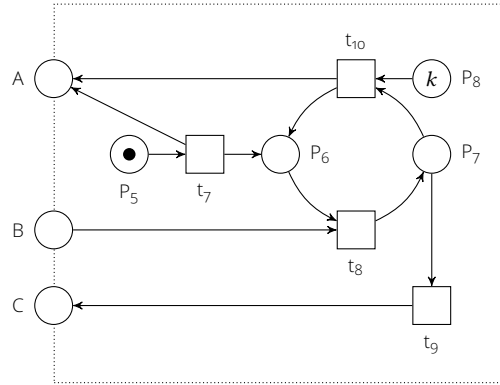


Abbildung 41 Ein Schema Net_{40} eines \mathbb{S} -Partners von Net_{36} , der für jedes $k \in \mathbb{N}_0$ den Zyklus in Net_{36} bis zu $k + 1$ -mal überdeckt. Jede Markierung von Net_{40} ist eine Zielmarkierung. Für ein bestimmtes $k \in \mathbb{N}_0$ schreiben wir Net_{36}^k für das entsprechende Netz.

deckt für alle $1 \leq \ell \leq k$ jeder Ablauf w von $\text{Net}_{41}^k \oplus \text{Env}(\text{Net}_{41}^k)$ mit $w|_{\text{Net}_{41}^k} = t_7 t_8 (t_{10} t_8)^\ell$ den Net_{36} -Zyklus z genau ℓ -mal.

^aTechnisch gesehen fehlt ein Stück Verhalten, das mit Net_{36} jedoch nicht eintreten kann: Nach Eintreten der Transition t_9 sowie in den Markierungen $\{P_5\}, \{P_6\}, \{P_7\}$ muss ein most-permissive Partner in der Lage dazu sein, ein B zu empfangen. Dadurch gelangt er in einen Zustand, der mit Net_{36} niemals eintreten kann und in dem jedes Verhalten erlaubt ist. Der Einfachheit halber haben wir dieses Verhalten nicht mit abgebildet.

Sei Z die Menge der L -Zyklen. Dann induziert R eine Funktion $Z \rightarrow \mathbb{N}_0 \cup \{\omega\}$, die jeden L -Zyklus z auf das Supremum $R[z]$ der Menge $\{w[z] \mid w \in \text{RUNS}(R \oplus \text{Env}(R))\}$ abbildet.

Beispiel 193

Wie in Beispiel 192, S. 149 betrachten wir das Netz Net_{36} aus Abbildung 36, S. 147 und seinen most-permissive Partner Net_{38} aus Abbildung 38, S. 148 sowie zwei weitere Partner Net_{40} aus Abbildung 40, S. 149 und Net_{41} aus Abbildung 41.

Net_{38} induziert die konstante Funktion $\text{Net}_{38}[z] = \omega$ für alle Net_{36} -Zyklen z . Sei z der Net_{36} -Zyklus $t_{10} t_{11} t_{12} t_8'$. Dann gilt $\text{Net}_{40}[z] = 0$ und $\text{Net}_{41}^k[z] = k$ für alle $k \in \mathbb{N}_0$.

Überdeckt ein Partner R einen L -Zyklus z nur endlich oft, gilt also $R[z] \in \mathbb{N}_0$, können wir alle Überdeckungen entfernen und erhalten einen Partner R' , der

z nicht mehr überdeckt. Außerdem überdeckt R' jeden L -Zyklus höchstens so häufig wie R , das heißt $R'[z'] \leq R[z']$ für alle $z' \in Z$.

Lemma 194

Sei L ein Netz und R ein \mathbb{S} -Partner von L . Sei $z \in Z$ und $R[z] \in \mathbb{N}_0$. Dann existiert ein \mathbb{S} -Partner R' mit $R'[z] = 0$ und $R'[z'] \leq R[z']$ für alle $z' \in Z$.

Beweis (Lemma 194)

O.B.d.A. nehmen wir im Folgenden an, dass die internen Plätze von R in $R \oplus \text{Env}(R)$ 1-beschränkt sind. Dies ist möglich, da wir $R \oplus \text{Env}(R)$ zu einem (möglicherweise unendlichen) Zustandsautomaten entfalten können.

Sei $z = \langle p, r \rangle$ und $R[z] = k$. Falls $k = 0$ gilt, gilt das Lemma trivial. Sei also im Folgenden $k > 0$. Sei W die Menge der Tupel $\langle w, q, q', y_1 \rangle$ mit

- (1) w ist ein Ablauf von $R \oplus \text{Env}(R)$ mit $w[z] > 0$,
- (2) w hat die Form $\overset{x}{\rightarrow} q \xrightarrow{ty} q' \xrightarrow{z}$, wobei w den L -Zyklus z mit ty überdeckt, also $\text{MARKINGS}_L^R(q) = \text{MARKINGS}_L^R(q')$ gilt. Dabei sei t eine Transition.

Sei $\langle w, q, q', t \rangle \in W$. Aus $k \in \mathbb{N}_0$ können wir trotz der möglichen Unendlichkeit von R folgern:

- (1) q ist *nicht* aus q' in $R \oplus \text{Env}(R)$ erreichbar.
- (2) Von q' aus sind nur endlich viele \hat{q} aus $\{\hat{q} \mid \langle \hat{w}, \hat{q}, \hat{q}', \hat{t} \rangle \in W\}$ in Reihe erreichbar.

Sei \underline{R} das Netz, das aus R wie folgt entsteht: Sei $u = \langle w, q, q', t \rangle \in W$. Dann fügen wir eine Transition \underline{u} ein, sodass $q \xrightarrow{\underline{u}} q'$ in $\underline{R} \oplus \text{Env}(\underline{R})$ gilt. Wir können leicht beobachten, dass \underline{R} ein \mathbb{S} -Partner von L ist, da die Erreichbarkeitsrelation in $L \oplus R$ und $L \oplus \underline{R}$ gleich ist.

Sei R' das durch Entfernen aller Transitionen t mit $\langle w, q, q', t \rangle \in W$ aus \underline{R} entstehende Netz. Wir zeigen, dass R' ein Partner von L ist. Dazu ist zu zeigen, dass $L \oplus R'$ schwach terminierend ist. Dazu müssen wir zweierlei zeigen:

- (1) Wir zeigen, dass jede erreichbare Markierung m von $L \oplus R'$ eine Rahmenmarkierung ist. Offenbar ist m eine erreichbare Markierung von $L \oplus \underline{R}$, da wir von \underline{R} nach R' nur Transitionen entfernt haben. Da der Schritt von R zu \underline{R} die Erreichbarkeitsrelation erhält, ist m also auch in $L \oplus R$ erreichbar. Da R ein \mathbb{S} -Partner ist, ist m also eine Rahmenmarkierung.

(2) Wir zeigen, dass aus jeder erreichbaren Markierung m von $L \oplus R'$ eine Zielmarkierung erreichbar ist. Wie bereits gezeigt ist m eine erreichbare Markierung von $L \oplus \underline{R}$ und $L \oplus R$. Daher gilt in $L \oplus R$: $\rightarrow m \xrightarrow{r} m'$ und m' ist eine Zielmarkierung. Wir unterscheiden zwei Fälle.

- a) r enthält *kein* t mit $\langle w, q, q', t \rangle \in W$. Dann gilt offenbar $m \xrightarrow{r} m'$ in $L \oplus R'$.
- b) r enthält ein t mit $u = \langle w, q, q', t \rangle \in W$. Also hat r die Form $r_1 t r_2$. Dann resultiert r_1 in einer Markierung m'' , in der t aktiviert ist. Also ist \underline{u} in m'' aktiviert und $r_1 \underline{u}$ ist ein Ablauf von $L \oplus R'$. Da jede erreichbare Markierung von $L \oplus R'$ auch eine erreichbare Markierung von $L \oplus R$ ist, ist m'' eine erreichbare Markierung von $L \oplus R$. Somit existiert ein s mit $m'' \xrightarrow{s} m'''$ in $L \oplus R$ wobei m''' eine Zielmarkierung ist. Da wie vorher gezeigt nur endlich viele \hat{q} in Reihe erreichbar sind mit $\hat{u} = \langle \hat{w}, \hat{q}, \hat{q}', \hat{t} \rangle \in W$, können wir nur endlich oft eine Transition \hat{t} in $L \oplus R$ in Reihe schalten. Also enthält s nur endlich viele solche \hat{t} . Also können wir durch endlich viele Ersetzungen der Form $\hat{t} \mapsto \hat{u}$ in s eine Verlängerung s' für $r_1 \underline{u}$ konstruieren, sodass $r_1 \underline{u} s'$ in der Zielmarkierung m''' resultiert.

Wir beobachten $R'[z'] \leq R[z']$ für alle $z' \in Z$, da die eingefügten Transitionen keine neuen Zyklen einführen können. Außerdem haben wir jede Überdeckung von z in R durch eine einzelne Transition ersetzt. Also gilt $R'[z] = 0$.

Ein \mathbb{S} -Partner R von L heißt *L-zyklus-minimal*, falls kein \mathbb{S} -Partner R' existiert mit

- (1) $R'[z] \leq R[z]$ für alle $z \in Z$.
- (2) $R'[z] < R[z]$ für wenigstens ein $z \in Z$.

Beispiel 195

Wie in Beispiel 193, S. 150 betrachten wir das Netz Net_{36} aus Abbildung 36, S. 147 und seinen most-permissive Partner Net_{38} aus Abbildung 38, S. 148 sowie zwei weitere Partner Net_{40} aus Abbildung 40, S. 149 und Net_{41} aus Abbildung 41, S. 150.

Das Netz Net_{40} ist Net_{36} -zyklus-minimal. Für $k > 0$ ist das Netz Net_{41} nicht Net_{36} -zyklus-minimal.

Mit Hilfe von Lemma 194, S. 151 können wir zeigen:

Satz 196 (Zyklenminimalität)

Sei R L -zyklus-minimal. Dann gilt für alle $z \in Z$ mit $R[z] \in \mathbb{N}_0$: $R[z] = 0$.

Beweis (Satz 196)

Folgt direkt aus Lemma 194, S. 151: Da R L -zyklus-minimal ist, existiert kein \mathbb{S} -Partner R' mit

- (1) $R'[z] \leq R[z]$ für alle $z \in Z$.
- (2) $R'[z] < R[z]$ für wenigstens ein $z \in Z$.

Gelte $0 \leq R[z] < \omega$. Dann existiert ein \mathbb{S} -Partner R' mit $R'[z] = 0$ und $R'[z'] \leq R[z']$ für alle $z' \in Z$. Also gilt $R[z] = 0$.

Wir können jedem L -Zyklus eine Menge von Zyklen in $L \oplus \text{MPP}(L)$ zuordnen. Gleichmaßen können wir jeden Zyklus in $L \oplus \text{MPP}(L)$ einen eindeutigen L -Zyklus zuordnen. Es existiert also eine Injektion i aus der Menge der Zyklen von $L \oplus \text{MPP}(L)$ in die Menge Z der L -Zyklen.

Gleichmaßen können wir jedem Zyklus z der Komposition $L \oplus R$ von L mit einem \mathbb{S} -Partner R einen äquivalenten Zyklus $j(z)$ aus $L \oplus \text{MPP}(L)$ zuordnen, da $\text{MPP}(L)$ most-permissive ist und daher $L \oplus R$ von $L \oplus \text{MPP}(L)$ simuliert wird (siehe Behauptung 180, S. 139). Auch j ist dabei eine Injektion.

Die Zuordnungen i und j können wir wie folgt zusammenführen, falls in $L \oplus R$ nur endlich viele Markierungen erreichbar sind: Sei x ein Zyklus von $L \oplus R$. Dann überdeckt R den L -Zyklus $i(j(x))$ *unbeschränkt*. Die Rückrichtung können wir ebenso zeigen: Sei x ein von R *unbeschränkt* überdeckter L -Zyklus. Sei y ein Zyklus von $L \oplus \text{MPP}(L)$ mit $i(y) = x$. Dann existiert in $L \oplus R$ ein Zyklus z mit $j(z) = y$, also $x = i(j(z))$.

Daraus können wir schließen: Sei x ein Zyklus von $L \oplus \text{MPP}(L)$. Existiere ein \mathbb{S} -Partner R , der $i(x)$ *beschränkt* überdeckt, und sei $L \oplus R$ beschränkt. Dann existiert ein \mathbb{S} -Partner $L \oplus R'$, sodass für alle Zyklen y von $L \oplus R$ gilt: $j(y) \neq x$.

7.3 Optimale Partner offener Systeme

Eine Präferenz \succsim auf offenen Systemen induziert eine Präferenz \succsim_L in Abhängigkeit von L : Zwei offene Systeme R und R' stehen in Relation, falls die jeweiligen Komposita $L \oplus R$ und $L \oplus R'$ in Relation stehen.

Definition 197 (Präferenz in Abhängigkeit eines offenen Systems)

Sei L ein Netz. Sei \succsim eine Präferenz. Dann ist \succsim_L die Präferenz mit $R \succsim_L R'$ g.d.w. $L \oplus R \succsim L \oplus R'$.

Häufig interessieren wir uns für die *optimalen Elemente* einer Teilmenge offener Systeme bezüglich einer Präferenz \succsim . Die maximalen Elemente einer Menge von Partnern von L bezüglich der Präferenz \succsim_L sind die *optimalen Partner*.

Definition 198 (Optimal, optimaler Partner)

Sei L ein Netz. Sei \mathcal{S} eine Menge von Netzen. Sei ρ eine Anforderung und \succsim eine Präferenz.

- (1) Sei $S \in \mathcal{S}$, sodass für alle $S' \in \mathcal{S}$ gilt: $S' \succsim S$ impliziert $S \succsim S'$. Dann ist S \succsim -*optimal* in der Menge \mathcal{S} .
- (2) Sei $R \in \rho[L]$ und \succsim_L -optimal in der Menge $\rho[L]$. Dann ist R ein \succsim -*optimaler ρ -Partner* von L .

Beispiel 199

Wie in Beispiel 182, S. 140 betrachten wir das Netz Net_{29} aus Abbildung 29, S. 138 jeweils in Komposition mit den Netzen Net_{30} aus Abbildung 30, S. 139, Net_{32} aus Abbildung 32, S. 142, Net_{33} aus Abbildung 33, S. 142 und dem Schema Net_{34} aus Abbildung 34, S. 143 mit $k \in \mathbb{N}_0$.

- Für die Perspektive $\text{FILTER}[t_1, t_3]$ ist jeder \mathbb{S} -Partner von Net_{29} ein optimaler \mathbb{S} -Partner.
- Für die Perspektive $\text{FILTER}[t_1, t_2, t_3]$ sind Net_{32} und Net_{34} mit $k = 0$ optimale \mathbb{S} -Partner. Net_{33} und Net_{34} für $k > 0$ jedoch nicht.
- Betrachten wir die Variante Net_{29}' von Net_{29} und den Anwendungsfall U aus Beispiel 182, S. 140, stellen wir fest, dass die optimalen \mathbb{S} -Partner für $\text{AF}[U]$ die optimalen \mathbb{S} -Partner für $\text{FILTER}[t_1, t_2, t_3]$ sind.

Notation 200 (Menge der optimalen Partner)

Sei L ein Netz, \mathcal{S} eine Menge von Netzen, ρ eine Anforderung und \succsim eine Präferenz. Wir führen die folgenden Notationen ein:

- (1) $\text{OPT}(\mathcal{S}, \succsim)$ für die Menge der \succsim -optimalen Elemente von \mathcal{S} .
- (2) $\rho[L]^\succsim$ für die Menge der \succsim -optimalen ρ -Partner von L .

$\rho[L]^\succsim$

Aus Definition 198, S. 154 folgt direkt der folgende Zusammenhang zwischen optimalen Elementen, optimalen Partnern und der Präferenz INDIFF :

Korollar 201

Sei L ein Netz, ρ eine Anforderung und \succsim eine Präferenz. Dann gelten:

$$\text{OPT}(\rho[L], \succsim_L) = \rho[L]^\succsim \quad (85)$$

$$\rho[L]^\succsim \subseteq \rho[L]^{\text{INDIFF}} \quad (86)$$

$$\rho[L]^{\text{INDIFF}} = \rho[L] \quad (87)$$

Wir heben die Begriffe der Bedienbarkeit und Ersetzbarkeit aus Definition 185, S. 143 auf optimale Partner:

Definition 202 (Optimal bedienbar, ersetzt optimal)

Sei ρ eine Anforderung und \succsim eine Präferenz. Seien L und L' Netze.

- (1) Sei $\rho[L]^\succsim \neq \emptyset$. Dann ist L *optimal bedienbar* bezüglich \succsim und ρ .
- (2) Sei $\rho[L]^\succsim \subseteq \rho[L']^\succsim$. Dann sagen wir: L' *ersetzt L optimal* bezüglich \succsim und ρ .

In dieser Arbeit betrachten wir oft die von einer Perspektive und einem Verbands-Kostenmodell induzierte Präferenz *kostengünstiger*. Wir führen daher die folgende Sprechweise ein:

Notation 203 (Kostenoptimal)

Sei π eine Perspektive und μ ein Verbands-Kostenmodell. Wir schreiben auch *kostenoptimal* bezüglich π und μ anstatt \succsim_μ^π -optimal.

7.3.1 Existenz optimaler Partner

Wir beobachten, dass nicht alle bedienbaren offenen Systeme auch optimal bedienbar sind:

Beispiel 204

Sei die Anforderung ρ die Menge aller endlichen Netze. Sei die Präferenz \succsim wie folgt definiert: $S \succsim S'$ g.d.w. S wenigstens so viele Plätze enthält wie S' . Dann ist jedes Netz bedienbar, jedoch keines optimal bedienbar: Zu jedem ρ -Partner mit k Plätzen können wir einen Partner mit $k + 1$ Plätzen angeben.

Wir interessieren uns in dieser Arbeit vorrangig für die Präferenz *kostengünstiger* aus Definition 142, S. 91. Dabei betrachten wir meist eine Filter-Perspektive und ein Verbands-Kostenmodell μ . In diesem Falle können wir genauere Aussagen über die Existenz kostenoptimaler Partner treffen. Insbesondere können wir eine Klasse von Kostenmodellen identifizieren, sodass jedes bezüglich einer beliebigen Anforderung ρ bedienbare Netz auch kostenoptimal bedienbar bezüglich ρ ist.

Sei im Folgenden L ein gewichtetes Netz über \mathbb{K} , π eine Perspektive von L und μ ein Verbands-Kostenmodell über \mathbb{K} und sei \top das Supremum von \mathbb{K} bezüglich \leq_μ . Sei außerdem ρ eine beliebige Anforderung. Die Existenz eines \succ_μ^π -optimalen ρ -Partners ist nicht trivial. Ganz allgemein können wir zwischen den folgenden Fällen unterscheiden:

- (1) Jeder ρ -Partner ist \succ_μ^π -optimal.
- (2) Einige, jedoch nicht alle ρ -Partner sind \succ_μ^π -optimal.
- (3) Kein ρ -Partner ist \succ_μ^π -optimal.

Wir betrachten das Problem abhängig von der $\rho \cap \mathbb{B}_\mu^\pi$ -Bedienbarkeit von L in Abschnitt 7.3.1.1 und Abschnitt 7.3.1.2 und setzen die Ergebnisse in Abschnitt 7.3.1.3, S. 160 zusammen.

7.3.1.1 Nicht $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbare offene Systeme

Sei L nicht $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar. Dann gilt also für jeden ρ -Partner R von L :

$$\llbracket L \oplus R \rrbracket_\mu^\pi = \top . \quad (88)$$

Als direkte Folge ist jeder ρ -Partner von L optimal bezüglich \succ_μ^π und es gilt:

$$\rho[L] = \rho[L] \succ_\mu^\pi . \quad (89)$$

Daher fällt die Existenz eines \succ_μ^π -optimalen ρ -Partners mit der Existenz eines ρ -Partners zusammen.

Korollar 205

Sei $\rho \cap \mathbb{B}_\mu^\pi[L] = \emptyset$. Dann sind die folgenden Aussagen äquivalent:

- (1) $\rho[L] = \emptyset$.
- (2) $\rho[L] \succ_\mu^\pi = \emptyset$.

7.3.1.2 $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbare offene Systeme

Dieser Fall ist substanziell schwieriger als aus Abschnitt 7.3.1.1, S. 156. Da L $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar ist, existiert ein $k \neq \top$ mit

$$\rho \cap \mathbb{B}_\mu^\pi[k][L] \neq \emptyset . \quad (90)$$

Also existiert auch ein ρ -Partner R von L und ein $k \neq \top$ mit

$$\llbracket L \oplus R \rrbracket_\mu^\pi = k. \quad (91)$$

Sei \mathcal{R}_k die Menge aller \mathbb{S} -Partner R mit $\llbracket L \oplus R \rrbracket_\mu^\pi = k$. Da die Präferenz *kostengünstiger* zwei Netze mit gleichen Kosten nicht unterscheidet, können wir weiter beobachten, dass genau einer der beiden Fälle eintritt:

$$\mathcal{R}_k \cap \rho[L]^{\succ_\mu^\pi} = \mathcal{R}_k \quad \text{oder} \quad (92)$$

$$\mathcal{R}_k \cap \rho[L]^{\succ_\mu^\pi} = \emptyset , \quad (93)$$

das heißt, entweder jeder oder kein $R \in \mathcal{R}_k$ ist \succ_μ^π -optimal.

Gleichung (92) ist genau dann erfüllt, wenn für alle $k' < k$ gilt:

$$\rho \cap \mathbb{B}_\mu^\pi[k'][L] = \emptyset . \quad (94)$$

Wir führen die folgende Notation ein:

Notation 206 (Schranken eines Systems)

Wir schreiben $\text{BOUNDS}_\mu^\pi(L, \rho)$ für die Menge aller $k \neq \top \in \mathbb{K}$ mit:

- (1) $\mathcal{R}_k \neq \emptyset$, und
- (2) Gleichung (94) ist für alle $k' < k$ erfüllt.

Beispiel 207

Wir betrachten die Netze Net_{29} aus Abbildung 29, S. 138, Net_{42} aus Abbildung 42 und Net_{43} aus Abbildung 43. Sei $\pi = \text{FILTER}[t_1, t_2, t_3]$, $\pi' = \text{FILTER}[t_1, t_2]$ und μ das durch die Max-Plus-Algebra induzierte Kostenmodell. Dann ist Net_{29} kostenbeschränkt bedienbar bezüglich π und μ sowie π' und μ . Das Netz Net_{42} ist kostenbeschränkt bedienbar bezüglich π' und $\mu \times \mu$, jedoch nicht kostenbeschränkt bedienbar bezüglich π und $\mu \times \mu$. Zuletzt ist Net_{43} kostenbeschränkt bedienbar bezüglich π und $\mu \times \mu$ sowie π' und $\mu \times \mu$.

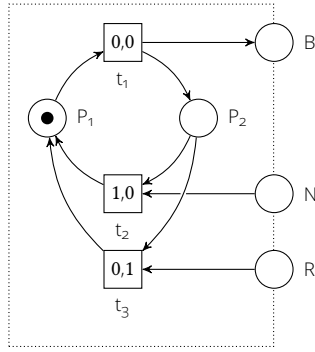


Abbildung 42 Eine Variante Net_{42} des Netzes Net_{29} aus Abbildung 29, S. 138, in der die Transitionen andere Kosten aus dem Kreuzprodukt $\mathbb{N}_0 \times \mathbb{N}_0$ aufweisen.

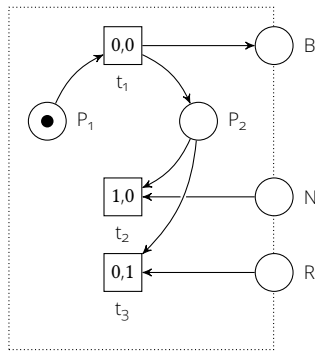


Abbildung 43 Eine Variante Net_{43} des Netzes Net_{42} aus Abbildung 42, in der immer nur eine Runde stattfindet. Die neue Zielmarkierung ist $\uparrow \downarrow$.

Wir beobachten

$$\begin{aligned} \text{BOUNDS}_{\mu}^{\pi}(\text{Net}_{29}, \mathbb{S}) &= \text{BOUNDS}_{\mu}^{\pi'}(\text{Net}_{29}, \mathbb{S}) = \{0\} \ , \\ \text{BOUNDS}_{\mu \times \mu}^{\pi}(\text{Net}_{43}, \mathbb{S}) &= \{\langle 0, 1 \rangle, \langle 1, 0 \rangle\} \ , \\ \text{BOUNDS}_{\mu \times \mu}^{\pi'}(\text{Net}_{42}, \mathbb{S}) &= \text{BOUNDS}_{\mu \times \mu}^{\pi'}(\text{Net}_{43}, \mathbb{S}) = \{\langle 0, 0 \rangle\} \ . \end{aligned}$$

Dann können wir schließen:

Korollar 208

Sei $\mathcal{R}_k \neq \emptyset$. Dann sind die folgenden Aussagen äquivalent:

- (1) $\rho[L]^{\succ_{\mu}^{\pi}} = \emptyset$.
- (2) $\text{BOUNDS}_{\mu}^{\pi}(L, \rho) = \emptyset$.

Ist \leq_μ total, können wir genauere Aussagen über die Anzahl der Elemente von $\text{BOUNDS}_\mu^\pi(L, \rho)$ treffen:

Lemma 209

Sei \leq_μ eine totale Ordnung. Wenn $\text{BOUNDS}_\mu^\pi(L, \rho)$ nicht-leer ist, dann ist $\text{BOUNDS}_\mu^\pi(L, \rho)$ einelementig.

Beweis (Lemma 209)

Sei $k \in \text{BOUNDS}_\mu^\pi(L, \rho)$, dann gilt für alle kleineren k' Gleichung (94), S. 157. Sei $\ell \in \text{BOUNDS}_\mu^\pi(L, \rho)$. Wir zeigen: $k = \ell$. Aus $\ell \in \text{BOUNDS}_\mu^\pi(L, \rho)$ folgt: $\mathcal{R}_k \neq \emptyset$. Also gilt $\ell \not\leq k$, da alle $k' < k$ Gleichung (94), S. 157 erfüllen. Da \leq_μ eine totale Ordnung ist, muss also $k \leq_\mu \ell$ gelten. Aus $\ell \in \text{BOUNDS}_\mu^\pi(L, \rho)$ erfüllt jedes ℓ' mit $\ell' < \ell$ Gleichung (94), S. 157. Da k Gleichung (94), S. 157 nicht erfüllt, gilt $k \not\leq \ell$ und $k \leq_\mu \ell$. Also gilt $k = \ell$.

Beispiel 210

Wie in Beispiel 207, S. 157 gezeigt, gilt

$$|\text{BOUNDS}_{\text{MP}}^{\text{FILTER}[t_1, t_2, t_3]}(\text{Net}_{29}, \mathbb{S})| = |\{0\}| = 1$$

für Net_{29} aus Abbildung 29, S. 138. Wir beobachten, dass die Forderung nach einer totalen Ordnung nur hinreichend, jedoch nicht notwendig ist: Es gilt auch

$$|\text{BOUNDS}_{\text{MP} \times \text{MP}}^{\text{FILTER}[t_1, t_2]}(\text{Net}_{42}, \mathbb{S})| = |\{\langle 0, 0 \rangle\}| = 1$$

für Net_{29} aus Abbildung 29, S. 138.

Wir betrachten im Folgenden den Spezialfall, dass \leq_μ eine Wohlordnung bis auf \top ist, also \leq_μ total ist und zu jeder nicht-leeren Menge $X \subseteq \mathbb{K}$ mit $\top \notin X$ ein kleinstes Element $x \in X$ von X bezüglich \leq_μ existiert. Dann können wir zeigen, dass L in jedem Falle optimal bedienbar bezüglich \succ_μ^π ist:

Lemma 211

Sei \leq_μ eine Wohlordnung bis auf \top . Sei $\rho \cap \mathbb{B}_\mu^\pi[L] \neq \emptyset$. Dann gelten:

- (1) $\text{BOUNDS}_\mu^\pi(L, \rho) \neq \emptyset$, und
- (2) $\rho[L]^{\succ_\mu^\pi} \neq \emptyset$.

Beweis (Lemma 211)

Wir zeigen Teil 1. Teil 2 folgt dann aus Korollar 208, S. 158. Da $L \cap \mathbb{B}_\mu^\pi$ -

bedienbar ist, existiert ein k , sodass L auch $\rho \cap \mathbb{B}_\mu^\pi[k]$ -bedienbar ist. Sei X die Menge der $k' < k$. Sei Y die Menge aller $k' \in X$, die Gleichung (94), S. 157 *nicht* erfüllen.

- (1) Sei Y leer. Dann erfüllen alle $k' < k$ Gleichung (94), S. 157. Also gilt $k \in \text{BOUNDS}_\mu^\pi(L, \rho)$.
- (2) Sei Y nicht leer. Da \leq_μ eine Wohlordnung bis auf \top ist, enthält Y ein kleinstes Element $y \in Y$. Da y ein kleinstes Element in Y ist, gilt für alle $k' < y$ die Gleichung (94), S. 157. Also gilt $y \in \text{BOUNDS}_\mu^\pi(L, \rho)$.

Kombinieren wir Lemma 209, S. 159 und Lemma 211, S. 159 erhalten wir:

Korollar 212

Sei \leq_μ eine Wohlordnung bis auf \top . Sei $\rho \cap \mathbb{B}_\mu^\pi[L] \neq \emptyset$. Dann ist die Menge $\text{BOUNDS}_\mu^\pi(L, \rho)$ einelementig.

Auf dieser Basis führen wir die folgende Schreibweise ein:

Notation 213 (Schranke unter Wohlordnung)

Sei \leq_μ eine Wohlordnung bis auf \top . Sei $\rho \cap \mathbb{B}_\mu^\pi[L] \neq \emptyset$. Dann schreiben wir $\text{BOUND}_\mu^\pi(L, \rho)$ für das einzige Element aus $\text{BOUNDS}_\mu^\pi(L, \rho)$.

Beispiel 214

Wie in Beispiel 207, S. 157 gezeigt, gilt

$$\text{BOUNDS}_{\text{MP}}^{\text{FILTER}[t_1, t_2, t_3]}(\text{Net}_{29}, \mathbb{S}) = \{0\}$$

für Net_{29} aus Abbildung 29, S. 138 und daher auch:

$$\text{BOUND}_{\text{MP}}^{\text{FILTER}[t_1, t_2, t_3]}(\text{Net}_{29}, \mathbb{S}) = 0 \quad .$$

7.3.1.3 Fazit

Wir die Ergebnisse aus Abschnitt 7.3.1.1, S. 156 und Abschnitt 7.3.1.2, S. 157 zusammen. Wenn \leq_μ eine Wohlordnung bis auf \top ist, können wir folgern, dass die Bedienbarkeit bezüglich ρ mit der optimalen Bedienbarkeit bezüglich \succsim_μ^π und ρ zusammenfällt:

Satz 215 (Koinzidenz: Bedienbarkeit, optimale Bedienbarkeit)

Sei \leq_μ eine Wohlordnung bis auf \top . Dann sind die folgenden Aussagen äquivalent:

- (1) $\rho[L] = \emptyset$.
- (2) $\rho[L]^{\succ_\mu^\pi} = \emptyset$.

Beweis (Satz 215)

Wir unterscheiden die folgenden zwei Fälle:

- (1) Sei L nicht $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar. Dann ist die Behauptung äquivalent zu Korollar 205, S. 156.
- (2) Sei L $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar. Dann ist L erst recht ρ -bedienbar, es gilt also $\rho[L] \neq \emptyset$. Aus Lemma 211, S. 159 folgt außerdem $\rho[L]^{\succ_\mu^\pi} \neq \emptyset$.

7.4 Diskussion

In diesem Abschnitt diskutieren wir die in diesem Kapitel getroffenen Entscheidungen zur Formalisierung sowie verwandte Ansätze.

7.4.1 Anwendung der Komposition auf mehr als zwei Operanden

Der in dieser Arbeit definierte Kompositionsoperator ist binär und hat die Eigenschaft der Kommutativität, es gilt also $L \oplus R = R \oplus L$. Bei der Komposition mehr als zweier offener Systeme $S_1 \dots S_n$ entsteht schnell die Forderung nach Assoziativität des Operators, das heißt, $S_1 \oplus (S_2 \oplus S_3) = (S_1 \oplus S_2) \oplus S_3$. Dann können wir nämlich die Komposition der offenen Systeme $S_1 \dots S_n$ einfach als $S_1 \oplus \dots \oplus S_n$ schreiben. Der in dieser Arbeit definierte Kompositionsoperator ist jedoch nicht assoziativ: Existiert ein Schnittstellenplatz p , sodass p in S_1 , S_2 und S_3 vorkommt, gilt $S_1 \oplus (S_2 \oplus S_3) \neq (S_1 \oplus S_2) \oplus S_3$, da wir stets nur bilaterale Kommunikation auf einem Puffer annehmen. Die folgenden Herangehensweisen sind denkbar, um für die hier definierte Komposition Assoziativität zu erreichen:

- (1) Es werden nur solche Mengen offener Systeme $S_1 \dots S_n$ komponiert für die gilt: Ist p ein Schnittstellenplatz eines S_i , dann existiert höchstens ein weiterer Index $j \neq i$, sodass p ein Schnittstellenplatz von S_j ist. Eine Menge, die mehr als zwei Indizes für p aufweist kann dann durch Umbenennen einzelner Plätze korrigiert werden. Diese Umbenennung ist nicht eindeutig und erfordert, genau festzulegen, welches System mit welchem kommunizieren soll.

- (2) Der Kompositionsoperator wird so abgeändert, dass die Kommunikation nicht mehr bilateral ist, das heißt, dass gemeinsame Schnittstellenplätze bei der Komposition nicht länger zu internen Plätzen werden wie bei Lohmann [Loh10b] sowie Dzikowski u. a. [Dzi+14]. Dadurch ist die Komposition zweier offener Systeme stets ein offenes System — und niemals geschlossen. Es wäre dann möglich einen Abschlussoperator einzuführen, der Schnittstellenplätze zu internen Plätzen macht.
- (3) Wir nehmen für jedes offene System eine in zwei Mengen partitionierte Schnittstelle an [Van+07; Rei09] Diese zwei Mengen bezeichnen wir als *linken Port* und *rechten Port*. Die Komposition $L \oplus R$ verbindet den rechten Port von L mit dem linken Port von R . Dabei werden überflüssige Schnittstellenplätze des rechten (linken) Ports von L (R) Teil des rechten (linken) Ports von $L \oplus R$. Dieser Ansatz verzichtet bewusst auf Kommutativität zu Gunsten der Assoziativität. Als Abschlussoperator könnten wir hier einen Operator auswählen, der die beiden eigenen Ports eines offenen Systems komponiert.

7.4.2 Vergleich mit anderen Kompositionsoperatoren

Der in dieser Arbeit vorliegende Kompositionsoperator unterstellt das Kommunikationsmodell der asynchronen, bilateralen Kommunikation über ungeordnete, unbeschränkte Puffer. Andere Kommunikationsmodelle erfordern möglicherweise andere Kompositionsoperatoren. So können wir beispielsweise unseren Kompositionsoperator für ungepufferte, synchrone Kommunikation erweitern, indem wir nicht nur Schnittstellenplätze sondern auch Schnittstellentransitionen zulassen. Übersichtsarbeiten über Kommunikationsmodelle im Bereich der Service-Orientierung sind zum Beispiel [Men05; KPS06; Loh10a].

Darüber hinaus existieren im Forschungsbereich der formalen Sprachen zahlreiche Kompositionsoperatoren, die Mengenoperationen wie Durchschnitt, Vereinigung, Konkatenation oder Shuffle nachbilden. Der Kompositionsoperator ist wohl am stärksten noch mit dem Shuffle-Operator verwandt: Sind L_1 und L_2 zwei Sprachen über zwei disjunkten Alphabeten Σ_1 und Σ_2 , ist das Shuffle von L_1 und L_2 die Sprache $\{w \mid w|_{\Sigma_1} \in L_1, w|_{\Sigma_2} \in L_2\}$. Die Abläufe von $L \oplus R$ können ebenfalls in ineinander verzahnte Sequenzen von Transitionen der beiden Komponenten zerlegt werden. Während Shuffle jedoch beliebige Verzahnungen zulässt, sind in unserem Fall nicht alle Verzahnungen erlaubt: Ob eine Verzahnung erlaubt ist, hängt einerseits vom internen Zustand der einzelnen Komponente sowie vom Inhalt der Puffer ab. Der Zusammenhang lässt sich in etwa wie folgt beschreiben: $\text{RUNS}(L \oplus R)$ ist eine Teilmenge des Shuffles der Sprachen $\{w|_L \mid w \in \text{RUNS}(L \oplus \text{ENV}(L))\}$ und $\{w|_R \mid w \in \text{RUNS}(L \oplus \text{ENV}(R))\}$. Komponieren wir zwei offene Systeme mit disjunkten Schnittstellen fallen

unsere Komposition und Shuffle der Mengen $\text{RUNS}(L)$ und $\text{RUNS}(R)$ zusammen. Betrachten wir anstatt asynchroner Kommunikation nur synchrone Kommunikation, entsteht ein Bezug zum Durchschnitt-Operator.

7.4.3 Partner für Anforderungen und Präferenzen

In dieser Arbeit betrachten wir Anforderungen und Präferenzen auf Komposita, um den Begriff des Partners und des optimalen Partners zu definieren, das heißt, L und R sind Partner, wenn das Kompositum $L \oplus R$ eine bestimmte Eigenschaft aufweist. Indem nur Partner komponiert werden, können wir also sicherstellen, dass jedes System korrekt ist. Das bedeutet gleichzeitig, dass ein System L von manchen Systemen ein Partner sein kann, während es von anderen Systemen kein Partner ist. Wir können also mithilfe des Partner-Begriffs ein einzelnes Netz L nicht als korrekt oder inkorrekt einstufen. Die Bedienbarkeit quantifiziert dieses Problem existenziell und wir können sie daher als ein notwendiges Kriterium für die Korrektheit ansehen: Ist L bedienbar, existiert ein R , sodass L und R Partner sind. Als ein hinreichendes Kriterium für die Korrektheit könnten wir anstatt der Existenz die Universalität betrachten: Ist L ein Partner *jedes* Systems R , interagiert L mit jedem anderen System zusammen korrekt. Dies ist beispielsweise trivial bei einer Anforderung der Fall, die von jedem System erfüllt wird. Bei einer solchen Anforderung ist jedoch jedes System korrekt und daher eine weitere Betrachtung unnötig. Für die in dieser Arbeit betrachteten Anforderungen können wir jeweils zeigen, dass ein L existiert, das nicht bedienbar ist. Also existiert für jedes L wenigstens ein R , sodass L und R keine Partner sind. Es ist denkbar, Anforderungen direkt an offene Systeme zu stellen, tatsächlich können wir die Partnerschaft mit einem bestimmten L als eine Anforderung an R verstehen. Betrachtet man Korrektheit einzelner Komponenten, ist es besonders wünschenswert, *kompositionale* Anforderungen zu identifizieren, das heißt Anforderungen für die gilt: Erfüllen L und R jeweils die Anforderung, sind L und R auch Partner. Die Anforderungen in dieser Arbeit sind jedoch nicht kompositional. Selbst im Falle kompositionaler Anforderungen ist die Korrektheit der einzelnen Komponenten zwar hinreichend, jedoch nicht notwendig. Daher ist ein solcher Ansatz möglicherweise zu streng.

8 Analyse- und Syntheseprobleme

In diesem Abschnitt formalisieren wir Analyse- und Syntheseprobleme, die wir in Abschnitt 1.2, S. 5 abstrakt formuliert haben. Wir beschreiben Lösungen für diese Probleme in Kapitel 10, S. 181 und Kapitel 12, S. 195, wobei wir jeweils die Konzepte aus Kapitel 9, S. 167 und Kapitel 11, S. 189 anwenden.

Sei L ein offenes, gewichtetes Netz über \mathbb{K} . Sei π eine Perspektive von L und μ ein Verbands-Kostenmodell über \mathbb{K} . Wie in Abschnitt 1.2, S. 5 motiviert, betrachten wir die folgenden Entscheidungs- und Syntheseprobleme für offene Systeme für $\rho \in \{\$, \mathbb{B}_\mu^\pi, \mathbb{B}_\mu^\pi[k], \$ \cap \mathbb{B}_\mu^\pi[k], \$ \cap \mathbb{B}_\mu^\pi\}$, $\succsim \in \{\succsim_\mu^\pi, \text{INDIFF}\}$ und eine Schwelle $k \neq \omega \in \mathbb{K}$:

Bedienbarkeit Entscheide ρ -Bedienbarkeit.

Partnersynthese Synthetisiere einen \succsim -optimalen ρ -Partner.

Repräsentation Berechne eine endliche Repräsentation aller \succsim -optimalen ρ -Partner.

Wir präzisieren das Problem der Repräsentation. Eine *Repräsentation* $\langle \bar{N}, \sim \rangle$ einer Klasse \mathcal{N} von Netzen besteht aus einem endlichen Netz \bar{N} und einer entscheidbaren, binären Relation \sim auf Netzen, sodass für jedes Netz N gilt: $\bar{N} \sim N$ g.d.w. $N \in \mathcal{N}$.

Wir präzentieren in Kapitel 10, S. 181 und Kapitel 12, S. 195 Lösungen für diese Probleme. Dabei beschränken wir uns ausdrücklich auf die folgende Netzklasse: Sei $b \geq 1$ eine für den Rahmen dieser Arbeit fest gewählte natürliche Zahl. Wir nehmen für jedes zu analysierende Netz L an, dass für jeden Platz p von L in jeder Rahmenmarkierung m die Ungleichung $m(p) \leq b$ gilt. Die konkrete Wahl von b ist dabei unerheblich, muss jedoch bekannt sein. Die Festlegung von b ist für unsere Zwecke notwendig, da ansonsten bereits die $\$$ -Bedienbarkeit unentscheidbar ist [Mas+08].

Für $\rho = \$$ und $\succsim = \text{INDIFF}$ wurden alle drei Probleme — Bedienbarkeit, Partnersynthese und Repräsentation — bereits gelöst. Das Ziel dieser Arbeit besteht darin, die Probleme für die restlichen Kombinationen aus ρ und \succsim auf die bereits gelösten Probleme zu reduzieren. Zu diesem Zwecke definieren wir *Repräsentanten* (siehe Kapitel 9, S. 167) und *Kostendiskriminanten* (siehe Kapitel 11, S. 189).

9 Repräsentanten offener Systeme

In diesem Kapitel betrachten wir die für Analyse und Synthese nützlichen *Repräsentanten offener Systeme*. Wir betrachten unterschiedliche Arten von Repräsentanten, je nach ihrer Anwendung in der Analyse. Erstens betrachten wir kompositionale Repräsentanten in Abschnitt 9.1, die Spezialfälle der Repräsentanten geschlossener Systeme aus Kapitel 4, S. 101 sind. Zweitens betrachten wir Partner-Repräsentanten in Abschnitt 9.2, S. 173, die einen Teil der in Kapitel 8, S. 165 motivierten Reduktionen ermöglichen. Schließlich diskutieren wir die Ergebnisse dieses Kapitels in Abschnitt 9.3, S. 179.

9.1 Kompositionale Repräsentanten

In Kapitel 4, S. 101 betrachteten wir Repräsentanten geschlossener Systeme. Einige Repräsentanten sind *kompositional*: Ist $L \oplus R$ ein Repräsentant für alle R , ist L ein kompositionaler Repräsentant. In den folgenden Abschnitten betrachten wir kompositionale Perspektiven-Repräsentanten (siehe Abschnitt 9.1.1), Wert-Repräsentanten (siehe Abschnitt 9.1.2, S. 170) und Schranken-Repräsentanten (siehe Abschnitt 9.1.3, S. 173).

9.1.1 Kompositionale Perspektiven-Repräsentanten

Sei L ein gewichtetes Netz über \mathbb{K} , seien π, π' Perspektiven von L . Sei μ ein Kostenmodell über \mathbb{K} .

In Abschnitt 4.1, S. 101 führten wir Perspektiven-Repräsentanten ein: Ein π, π' -Perspektiven-Repräsentant bezüglich μ unterscheidet nicht zwischen den beiden Perspektiven π und π' . Bei einem *kompositionalen Perspektiven-Repräsentanten* setzt sich diese Eigenschaft auf die Komposition mit beliebigen Netzen fort.

Definition 216 (Kompositionaler Perspektiven-Repräsentant)

Gelte für alle R : $L \oplus R$ ist ein π, π' -Perspektiven-Repräsentant bezüglich μ . Dann ist L ein *kompositionaler π, π' -Perspektiven-Repräsentant* bezüglich μ .

Nicht jeder π, π' -Perspektiven-Repräsentant ist kompositional: Bei der Komposition mit einem Netz R kann in L neues Verhalten auftreten. Betrachten wir dazu Net_{29} aus Abbildung 29, S. 138. Offenbar ist Net_{29} ein $\text{ID}, \text{FILTER}[\{t_1\}]$ -Repräsentant, denn die Transitionen t_2 und t_3 können niemals schalten. Wir betrachten zum Vergleich die Komposition $\text{Net}_{31} = \text{Net}_{29} \oplus \text{Net}_{30}$ aus Abbildung 31, S. 139. Wir beobachten, dass die Transitionen t_2 und t_3 in Net_{31} schalten können. Ist μ durch die Max-Plus-Algebra induziert, gilt $\llbracket \text{Net}_{13} \rrbracket_{\mu}^{\text{ID}} = \omega$ und $\llbracket \text{Net}_{13} \rrbracket_{\mu}^{\text{FILTER}[t_1]} = 0$. Folglich ist Net_{29} *kein kompositionaler* $\text{ID}, \text{FILTER}[t_1]$ -Perspektiven-Repräsentant.

Wir betrachten in den folgenden Unterabschnitten die Kompositionalität der Konstruktionen aus Abschnitt 4.1, S. 101. Die Ergebnisse fassen wir in Abbildung 44 im Stil von Abbildung 23, S. 103 zusammen.

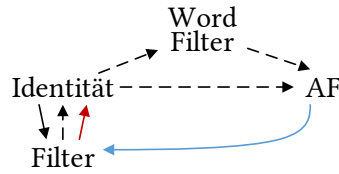


Abbildung 44 Konditionale Existenz kompositionaler Perspektiven-Repräsentanten (gestrichelt), teils nur für Verbands-Kostenmodelle (blau) oder die Max-Plus-Algebra als Kostenmodell (rot)

9.1.1.1 Von der Identität zu Filter-Perspektiven

Sei L ein Netz mit Transitionen T . Sei π die Perspektive ID von L . Dann ist $L \oplus R$ stets ein $\pi, \text{FILTER}[T]$ -Perspektiven-Repräsentant: $\pi(L \oplus R)$ beschreibt per Konvention $\text{FILTER}[T] \triangleright \pi(L \oplus R) = \text{FILTER}[T] \triangleright \text{FILTER}[T](L \oplus R)$. Also ist L ein kompositionaler Perspektiven-Repräsentant.

9.1.1.2 Wechsel zwischen Identität und Terminierungs-Perspektive

Sei L ein Netz mit Transitionen T . Sei π die Perspektive ID von L . Wir beobachten, dass L nur dann ein kompositionaler $\pi, \text{TERM}[N]$ -Perspektiven-Repräsentant ist, falls die Kosten von $L \oplus R$ für beliebige R gleich sind. Wir könnten nämlich ein R mit einer leeren Menge von Zielmarkierungen wählen. Die in Abschnitt 4.1.2, S. 104 vorgeschlagene Konstruktion ist deshalb nicht auf Komposita übertragbar, weil die Zielmarkierungen sowohl in L wie in R angepasst werden müssten. Gleichmaßen lässt sich auch die zweite Idee aus Abschnitt 4.1.2, S. 104 nicht anwenden: Nicht jedes Kompositum $L \oplus R$ ist schwach terminierend.

9.1.1.3 Von der Terminierungs-Perspektive zur Filter-Perspektive

Die in Abschnitt 4.1.2, S. 104 vorgestellte Konstruktion resultiert deshalb nicht in einem kostenäquivalenten, kompositionalen $\text{TERM}[L]$, $\text{FILTER}[T]$ -Perspektiven-Repräsentanten, weil die Zielmarkierungen von $L \oplus R$ stets auch von R abhängen.

9.1.1.4 Von der Identität zu Anwendungsfall-Perspektiven

Sei L ein beschriftetes Netz über Σ mit Transitionen T . Komme ε nicht als Beschriftung in L vor. Sei π die Perspektive ID von L . Sei U der Anwendungsfall mit $\bullet U = \{\varepsilon\}$ und $U^\bullet = \Sigma^*$. Die Perspektive $\pi(L \oplus R)$ beschreibt per Konvention $\text{FILTER}[T] \triangleright \pi(L \oplus R)$. Die Perspektive $\text{AF}[L, U](L \oplus R)$ beschreibt per Konvention $\text{FILTER}[T] \triangleright \text{AF}[L, U](L \oplus R)$. Der Anwendungsfall U ist so gewählt, dass jeder Ablauf ein Teil des U -Verhaltens ist. Daher ist $L \oplus R$ stets ein π , $\text{AF}[L, U]$ -Perspektiven-Repräsentant und somit L stets ein kompositionaler π , $\text{AF}[L, U]$ -Perspektiven-Repräsentant.

9.1.1.5 Von Anwendungsfall-Perspektiven zu Filter-Perspektiven

Wir zeigen, dass die Konstruktion aus Abschnitt 4.1.5, S. 106 stets einen kompositionalen Perspektiven-Repräsentanten liefert. Sei dazu L ein Netz mit Transitionen T und U ein Anwendungsfall von L . Sei R ein Netz. Wir unterscheiden die folgenden zwei Fälle:

- (1) In R kommt keine Beschriftung aus L vor. Dann gilt $L_U \oplus R = (L \oplus R)_U$, da nur Transitionen von L mit Transitionen aus dem U -erkennenden Automaten synchronisiert werden. Folglich bleibt auch die Menge $T \times W$ gleich. Daher werden alle neu eingefügten Plätze und Transitionen nur mit L verbunden. Daraus folgern wir, dass die Konstruktion sowohl vor als auch nach der Komposition vorgenommen werden kann.
- (2) In R existiert wenigstens eine Transition mit einer Beschriftung, die in L vorkommt. Dann ist die Gleichheit $L_U \oplus R = (L \oplus R)_U$ nicht mehr gegeben. Wir benötigen daher ein weiteres Argument: Per Konvention steht $\text{AF}[L, U](L \oplus R)$ für $\text{FILTER}[T] \triangleright \text{AF}[L, U](L \oplus R)$. Von daher sind die Transitionen von R nicht für die Abdeckung des Anwendungsfalls relevant.

9.1.1.6 Von der Identität zu Wort-Filter-Perspektiven

Sei L ein Netz mit Transitionen T und habe jede Transition $t \in T$ die Beschriftung ε . Sei π die Perspektive ID von L . Dann ist L ein π , $\text{WFILTER}[L, \varepsilon]$ -Perspektiven-Repräsentant. Per Konvention steht $\pi(L \oplus R)$ für $\text{FILTER}[T] \triangleright \text{ID}(L \oplus R)$ und $\text{WFILTER}[L, \varepsilon](L \oplus R)$ für $\text{FILTER}[T] \triangleright \text{WFILTER}[L, \varepsilon](L \oplus R)$. Da jede Transition in L mit ε beschriftet ist, gilt $\text{FILTER}[T] \triangleright \text{ID}(L \oplus R) = \text{FILTER}[T] \triangleright \text{WFILTER}[L, \varepsilon](L \oplus R)$. Folglich ist L ein kompositionaler π , $\text{WFILTER}[L, \varepsilon]$ -Perspektiven-Repräsentant.

9.1.1.7 Von Filter-Perspektiven zur Identitäts-Perspektive

Sei L ein Netz mit Transitionen T . Sei $T' \subseteq T$, sodass für jede Transition $t \in T \setminus T'$ und jedes Netz R gilt: t kann in $L \oplus R$ niemals schalten. Dann ist L ein kompositionaler $\text{FILTER}[T']$, ID -Perspektiven-Repräsentant. Wird das betrachtete Kostenmodell durch einen Semiring induziert, können wir die Kosten jeder Transition $t \in T \setminus T'$ auf $\mathbb{1}$ setzen. Das resultierende Netz L' ist ein kompositionaler $\text{FILTER}[T']$, ID -Perspektiven-Repräsentant.

9.1.1.8 Von Wort-Filter-Perspektiven zu Anwendungsfall-Perspektiven

Sei L ein Netz und die in Abschnitt 4.1.8, S. 109 geforderten Bedingungen erfüllt. Dann können wir auch hier den kanonischen Anwendungsfall U bilden mit $\bullet U = \{\varepsilon\}$ und $U^\bullet = \{w\}$. Folglich ist L bereits ein kompositionaler $\text{WFILTER}[L, w]$, $\text{AF}[L, U]$ -Perspektiven-Repräsentant, denn die Komposition kann die geforderten Bedingungen nicht verletzen, und zusätzlich verhindert die für L kanonische Filter-Perspektive in der Komposition $L \oplus R$, dass Transitionen von R die Erzeugung des Wortes w beeinflussen.

9.1.1.9 Von der Terminierungs-Perspektive zur Anwendungsfall-Perspektive

Analog zu Abschnitt 9.1.1.3, S. 169 ist es allgemein nicht möglich, das terminierende Verhalten von $L \oplus R$ für alle R in einem einzigen Anwendungsfall U von L zu beschreiben.

9.1.2 Kompositionale Wert-Repräsentanten

In Kapitel 4, S. 101 betrachteten wir Wert-Repräsentanten: Netze, deren Wertsemantik sich aus den erreichbaren Markierungen ableiten lässt. Dabei trägt der Begriff eine Funktion $f: \wp(P') \setminus \emptyset \rightarrow \mathbb{K}$ als Parameter, wobei P' eine

Teilmenge der Plätze ist. Die Wertsemantik eines Wert-Repräsentanten ist im Falle eines in P' beschränkten Netzes aus endlich vielen Abläufen und im Falle eines in P' unbeschränkten Netzes aus der Abbildung f ableitbar. Ein Wert-Repräsentant L ist kompositional, falls $L \oplus R$ ein Wert-Repräsentant für beliebige R ist.

Definition 217 (Kompositionaler Wert-Repräsentant)

Sei L ein gewichtetes Netz über \mathbb{K} mit Plätzen P . Sei $P' \subseteq P$ und $f: \wp(P') \setminus \emptyset \rightarrow \mathbb{K}$. Sei π eine Perspektive von L , sei μ ein Kostenmodell über \mathbb{K} . Gelte für alle Netze R : $L \oplus R$ ist ein f -Wert-Repräsentant bezüglich π und μ .

Dann ist L ein *kompositionaler f -Wert-Repräsentant* bezüglich π und μ .

Jeder kompositionale Wert-Repräsentant L ist auch stets ein Wert-Repräsentant, denn wir können stets das leere Netz für R wählen und es gilt $L \oplus R = L$. Die Umkehrung gilt im Allgemeinen jedoch nicht: Die Argumentation ist dabei analog zu der aus dem vorherigen Abschnitt 9.1.1, S. 167. Die Kosteneindeutigkeit bleibt in der Komposition im Allgemeinen nicht erhalten:

Beispiel 218

Betrachten wir dazu Net_{29} aus Abbildung 29, S. 138. Offenbar ist Net_{29} ein f -Repräsentant für beliebige f , da Net_{29} kosteneindeutig und beschränkt ist. Wir vergleichen mit der Komposition $\text{Net}_{31} = \text{Net}_{29} \oplus \text{Net}_{30}$ aus Abbildung 31, S. 139: Diese ist nicht kosteneindeutig bezüglich der Identität als Perspektive und dem aus der Max-Plus-Algebra induzierten Kostenmodell: Die Abläufe $t_1 t_4 t_5 t_2$ und $t_1 t_5 t_6 t_3$ enden in der gleichen Markierung, weisen jedoch unterschiedliche Kosten 1 und 0 auf. Folglich ist Net_{29} *kein kompositionaler f -Wert-Repräsentant* für beliebige f .

Die entwickelten Konstruktionsvorschriften aus Abschnitt 4.2.2, S. 112 für Wert-Repräsentanten liefern jedoch jeweils auch kompositionale Wert-Repräsentanten, wie wir folgend zeigen.

9.1.2.1 Kostenmodelle über Verbänden

Wir zeigen, dass die Konstruktion aus Abschnitt 4.2.1, S. 111 kompositional ist. Dafür zeigen wir, dass es unerheblich ist, ob die Konstruktion des Wert-Repräsentanten vor oder nach der Komposition stattfindet.

Lemma 219

Seien L, R Netze. Sei L gewichtet über \mathbb{K} . Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von L . Sei μ Verbands-Kostenmodell über \mathbb{K} . Sei G ein Graph von $\mathbb{K}[L, \pi]$ in μ .

Dann gilt: $\text{REP}_\mu^\pi(L, G) \oplus R = \text{REP}_\mu^\pi(L \oplus R, G)$.

Beweis (Lemma 219)

In der Konstruktion werden alle neu eingeführten Plätze einzig mit Transitionen aus T' verbunden und alle Transitionen von $A_{L, \mu}$ nur mit Transitionen aus T' synchronisiert. Folglich kann die Komposition vor oder nach der Konstruktion angewendet werden, ohne das Ergebnis zu beeinflussen.

Korollar 220

Sei L ein gewichtetes Netz über \mathbb{K} . Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von L . Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei G ein Graph von $\mathbb{K}[L, \pi]$ in μ .

Dann gilt: $\text{REP}_\mu^\pi(L, G)$ ist ein kompositionaler f -Wert-Repräsentant.

9.1.2.2 Das Kostenmodell der Max-Plus-Algebra

Wir zeigen, dass $\text{REP}_{\text{MP}}^{T'}(N)$ kompositional ist. Dazu zeigen wir, dass es unerheblich ist, ob die Konstruktion des Wert-Repräsentanten vor oder nach der Komposition stattfindet:

Lemma 221

Seien L, R Netze. $\text{REP}_{\text{MP}}^{T'}(L) \oplus R = \text{REP}_{\text{MP}}^{T'}(L \oplus R)$.

Beweis (Lemma 221)

In der Konstruktion von $\text{REP}_{\text{MP}}^{T'}(L)$ werden alle neu eingeführten Plätze einzig mit Transitionen aus T' verbunden. Daher kann die Komposition vor oder nach der Konstruktion angewendet werden, ohne das Ergebnis zu beeinflussen.

Daraus können wir schließen:

Korollar 222

$\text{REP}_{\text{MP}}^{T'}(N)$ ist ein kompositionaler $\{P \mapsto \omega \mid P \neq \emptyset \in \wp(\{\hat{p}, \hat{p}_{-\infty}\})\}$ -Wert-Repräsentant.

9.1.3 Kompositionale Schranken-Repräsentanten

Analog zu kompositionalen Wert-Repräsentanten definieren wir *kompositionale Schranken-Repräsentanten* auf Basis der Definition 156, S. 114:

Definition 223 (Kompositionaler Schranken-Repräsentant)

Sei L ein gewichtetes Netz über \mathbb{K} . Sei $k \in \mathbb{K}$ und p ein Platz von L . Sei μ ein Kostenmodell über \mathbb{K} und π eine Perspektive. Gelte für alle R : $L \oplus R$ ist ein p, k -Schranken-Repräsentant bzgl. π und μ . Dann ist L ein *kompositionaler p, k -Schranken-Repräsentant*.

Wie im vorherigen Abschnitt 9.1.2, S. 170 können wir zeigen, dass die in Abschnitt 4.3, S. 113 konstruierten Schranken-Repräsentanten kompositional sind. Dabei können wir dieselben Argumente wie in den jeweiligen Beweisen von Lemma 219, S. 172 und Lemma 221, S. 172 anwenden.

Korollar 224

Seien L, R Netze. Sei L gewichtet über \mathbb{K} . Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von L . Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ eine Schwelle. Sei G ein k -Graph von $\mathbb{K}[L, \pi]$ in μ .

Dann gilt: $\text{REP}_\mu^\pi[k](L, G) \oplus R = \text{REP}_\mu^\pi[k](L \oplus R, G)$.

Korollar 225

Sei L ein gewichtetes Netz über \mathbb{K} . Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von L . Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ eine Schwelle. Sei G ein Graph von $\mathbb{K}[L, \pi]$ in μ .

$\text{REP}_\mu^\pi[k](L, G)$ ist ein kompositionaler $\not\prec, k$ -Schranken-Repräsentant.

9.2 Partner-Repräsentanten

In diesem Abschnitt betrachten wir ein offenes System L sowie die Menge $\rho[L]^\succ$ seiner optimalen Partner. Für einige Anforderungen und Präferenzen existieren bereits Analysetechniken in der Literatur, jedoch nicht für die für uns interessanten Kombinationen aus Anforderung ρ und Präferenz \succ mit $\rho \in \{\mathbb{B}_\mu^\pi, \mathbb{B}_\mu^\pi[k], \mathbb{SB}_\mu^\pi, \mathbb{SB}_\mu^\pi[k]\}$ und $\succ \in \{\text{INDIFF}, \succ_\mu^\pi\}$.

Unser Ziel ist es, bestehende Techniken für unsere Probleme zu verwenden. Nehmen wir dazu an, dass wir Analysetechniken für $\langle \rho, \succ \rangle$ suchen. Seien Analysetechniken für das Paar $\langle \rho', \succ' \rangle$ aus Anforderung ρ' und Präferenz \succ' bekannt. Wir identifizieren einen wichtigen Spezialfall: Gilt $\rho[L]^\succ = \rho'[L]^\succ'$, können wir direkt die bekannten Techniken anwenden. Erfüllt L

diese Eigenschaft, nennen wir L einen *Partner-Repräsentanten* bezüglich $\rho, \rho', \succ, \succ'$. Ein *Partner-Repräsentant* L' von L repräsentiert also die Menge $\rho[L]^\succ$ durch die Menge $\rho'[L']^{\succ'}$ seiner eigenen optimalen Partner. Dabei darf sowohl die betrachtete Anforderung als auch die betrachtete Präferenz wechseln.

Wir zeigen folgend Konstruktionen vor, um zu einem gegebenen Netz L einen äquivalenten Partner-Repräsentanten L' zu erzeugen. Das heißt, dass die folgenden Gleichungen gelten:

$$\rho[L]^\succ = \rho[L']^\succ \quad (95)$$

$$= \rho'[L']^{\succ'} \quad (96)$$

9.2.1 Partner-Repräsentanten für beliebige Anforderungen und Präferenzen

Wir formalisieren, dass die betrachtete Anforderung und die betrachtete Präferenz wechseln, durch den Begriff der *Substitution*.

Definition 226 (Substitution)

Seien ρ_1, ρ_2 Anforderungen und \succ_1, \succ_2 Präferenzen.

Dann ist $\sigma = \langle \rho_1, \rho_2, \succ_1, \succ_2 \rangle$ eine *Substitution*.

Wir beobachten, dass wir zu jedem Paar von Anforderungen auch eine Substitution bilden können, indem wir für \succ_1 und \succ_2 jeweils die Präferenz wählen, die alle offenen Systeme in Relation setzt. Analog können wir immer die Anforderung annehmen, die jedes Netz erfüllt.

Notation 227 (Substitution)

Sei $\sigma = \langle \rho_1, \rho_2, \succ_1, \succ_2 \rangle$ eine Substitution. Wir führen die folgenden Schreibweisen für σ ein:

- (1) $\{\rho_1 \mapsto \rho_2, \succ_1 \mapsto \succ_2\}$,
- (2) $\{\rho_1 \mapsto \rho_2\}$, falls für alle Netze S, T gilt: $S \succ_1 T$ und $S \succ_2 T$.
- (3) $\{\succ_1 \mapsto \succ_2\}$, falls für alle Netze S gilt: S erfüllt ρ_1 und S erfüllt ρ_2 .

Basierend auf der Substitution formalisieren wir den Begriff des Partner-Repräsentanten.

Definition 228 (Partner-Repräsentant)

Sei $\sigma = \{\rho_1 \mapsto \rho_2, \succ_1 \mapsto \succ_2\}$ eine Substitution. Ein Netz L ist ein σ -Partner-Repräsentant, falls

$$\rho_1[L]^{\succ_1} = \rho_2[L']^{\succ_2} . \quad (97)$$

Partner-Repräsentant

Notation 229 (Mengen von Partner-Repräsentanten)

Sei L ein Netz und $\sigma = \{\rho_1 \mapsto \rho_2, \succ_1 \mapsto \succ_2\}$ eine Substitution. Wir schreiben $\text{REPS}_\sigma(L)$ für die Menge der σ -Partner-Repräsentanten L' mit $\rho_1[L]^{\succ_1} = \rho_1[L']^{\succ_1}$.

9.2.2 Partner-Repräsentanten für Kostenbeschränktheit

Sei L ein Netz über \mathbb{K} . Sei π eine Perspektive von L , die unabhängig von der Wahl der Rahmenmarkierungen von L ist. Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ und $\sigma = \{\mathbb{S}\mathbb{B}_\mu^\pi[k] \mapsto \mathbb{S}\}$.

Wir zeigen folgenden Zusammenhang zwischen Partner-Repräsentanten und kompositionalen Schranken-Repräsentanten: Ist L ein kompositionaler Schranken-Repräsentant, können wir L in einen σ -Partner-Repräsentanten L' aus der Menge $\text{REPS}_\sigma(L)$ überführen. Hierfür passen wir die Menge der Rahmenmarkierungen an, indem wir den Durchschnitt der Rahmenmarkierungen mit der Menge der Markierungen bilden, deren Kosten die Schranke k nicht überschreiten. Hiermit schränken wir die Menge der \mathbb{S} -Partner von L auf jene ein, die die Kostenschranke k einhalten. Die Menge der $\mathbb{B}_\mu^\pi[k]$ -Partner bleibt dabei jedoch unverändert.

Notation 230 (Anpassung der Rahmenmarkierungen)

Sei L ein p -Schranken-Repräsentant. Sei M^\square die Menge der Rahmenmarkierungen von L . Dann schreiben wir $L[\sigma]$ für die Variante von L mit der Menge $\{m \mid m \in M^\square, m(p) = 0\}$ von Rahmenmarkierungen.

Lemma 231

Sei L ein kompositionaler p, k -Schranken-Repräsentant. Dann gelten die folgenden Aussagen:

- (1) $L[\sigma]$ ist ein kompositionaler p, k -Schranken-Repräsentant.
- (2) $L[\sigma]$ ist ein σ -Partner-Repräsentant.
- (3) $L[\sigma] \in \text{REPS}_\sigma(L)$.

Beweis (Lemma 231, S. 175)

- (1) L ist ein kompositionaler p, k -Schranken-Repräsentant. Da π unabhängig von den Rahmenmarkierungen ist, gilt für alle Netze R : $\llbracket L \oplus R \rrbracket_\mu^\pi = \llbracket L[\sigma] \oplus R \rrbracket_\mu^\pi$.
- (2) Zu zeigen: $\mathbb{SB}_\mu^\pi[k][L[\sigma]] = \mathbb{S}[L[\sigma]]$.
- a) $\mathbb{SB}_\mu^\pi[k][L[\sigma]] \subseteq \mathbb{S}[L[\sigma]]$ folgt direkt aus $\mathbb{SB}_\mu^\pi[k] \subseteq \mathbb{S}$ und Lemma 184, S. 142.
- b) Sei $R \in \mathbb{S}[L[\sigma]]$. Dann gilt $L[\sigma] \oplus R \in \mathbb{S}$. Also ist jede erreichbare Markierung m von $L[\sigma] \oplus R$ eine Rahmenmarkierung. Also gilt laut Notation 230, S. 175 $m(p) = 0$. Da $L[\sigma]$ ein kompositionaler p, k -Schranken-Repräsentant ist, ist $L[\sigma] \oplus R$ ein p, k -Schranken-Repräsentant. In jeder erreichbaren Markierung gilt: $m(p) = 0$. Deshalb gilt $\llbracket L[\sigma] \oplus R \rrbracket_\mu^\pi \leq_\mu k$ und folglich auch $R \in \mathbb{B}_\mu^\pi[k][L[\sigma]]$. Nach Lemma 184, S. 142 gilt somit $R \in \mathbb{SB}_\mu^\pi[k][L[\sigma]]$.
- (3) Zu zeigen: $\mathbb{SB}_\mu^\pi[k][L] = \mathbb{SB}_\mu^\pi[k][L[\sigma]]$. Wir sehen, dass $\text{RUNS}(L \oplus R, m) = \text{RUNS}(L[\sigma] \oplus R, m)$ für alle R und Markierungen m gilt. Daraus folgt $\llbracket L \oplus R \rrbracket_\mu^\pi = \llbracket L[\sigma] \oplus R \rrbracket_\mu^\pi$ und daher auch $\mathbb{B}_\mu^\pi[k][L] = \mathbb{B}_\mu^\pi[k][L[\sigma]]$. Wir können außerdem $\mathbb{S}[L] \supseteq \mathbb{S}[L[\sigma]]$ beobachten, denn jede Rahmenmarkierung von $L[\sigma]$ ist auch eine Rahmenmarkierung von L . Wir folgern mit Lemma 184, S. 142 $\mathbb{SB}_\mu^\pi[k][L] \supseteq \mathbb{SB}_\mu^\pi[k][L[\sigma]]$. Sei $R \in \mathbb{SB}_\mu^\pi[k][L]$. Dann gilt für alle erreichbaren Markierungen m von $L \oplus R$: $\llbracket L \oplus R, m \rrbracket_\mu^\pi \leq_\mu k$ und m ist eine Rahmenmarkierung. Da L ein kompositionaler p, k -Schranken-Repräsentant ist, ist $L \oplus R$ ein p, k -Schranken-Repräsentant. Damit gilt $m(p) = 0$. Entsprechend ist m auch eine Rahmenmarkierung von $L[\sigma] \oplus R$. Folglich ist $L[\sigma] \oplus R$ schwach terminierend und damit gilt $R \in \mathbb{S}[L[\sigma]]$. Aus Lemma 184, S. 142 folgt $\mathbb{SB}_\mu^\pi[k][L] \subseteq \mathbb{SB}_\mu^\pi[k][L[\sigma]]$.

Jede Konstruktion, die L in einen kompositionalen Schranken-Repräsentanten L' mit gleicher Partnermenge überführt, lässt sich entsprechend leicht erweitern, sodass der Partner-Repräsentant $L'[\sigma]$ entsteht.

Wir zeigen, dass ein in Abschnitt 4.3.1, S. 114 konstruierter Schranken-Repräsentant für L in einem Partner-Repräsentanten $L'[\sigma]$ von L resultiert.

Satz 232 (Partner-Repräsentant aus Schranken-Repräsentant)

Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive von L . Sei G ein k -Graph von $\mathbb{K}[L, \pi]$ in μ .

Dann gilt: $\text{REP}_\mu^\pi[k](L, G)[\sigma] \in \text{REPS}_\sigma(L)$.

Beweis (Satz 232)

Wir zeigen, dass L und $\text{REP}_\mu^\pi[k](L, G)$ dieselbe Partnermenge bezüglich $\mathbb{SB}_\mu^\pi[k]$ aufweisen. Sei R ein Netz. Wir zeigen: R ist ein $\mathbb{SB}_\mu^\pi[k]$ -Partner von L g.d.w. R ein \mathbb{S} -Partner von $\text{REP}_\mu^\pi[k](L, G)$ ist. Korollar 224, S. 173 besagt: $\text{REP}_\mu^\pi[k](L, G) \oplus R = \text{REP}_\mu^\pi[k](L \oplus R, G)$. Laut Lemma 125, S. 85 sind $L \oplus R$ und $\text{REP}_\mu^\pi[k](L \oplus R, G)$ bisimilar. Also sind auch $\text{REP}_\mu^\pi[k](L, G) \oplus R$ und $L \oplus R$ bisimilar. Folglich sind $L \oplus R$ und $\text{REP}_\mu^\pi[k](L, G) \oplus R$ entweder beide schwach terminierend oder beide nicht schwach terminierend. Somit haben L und $\text{REP}_\mu^\pi[k](L, G)$ dieselben \mathbb{S} -Partner. Da $\text{REP}_\mu^\pi[k](L, G)$ ein kompositionaler und zu L kosten-äquivalenter Schranken-Repräsentant ist, haben $L \oplus R$ und $\text{REP}_\mu^\pi[k](L, G) \oplus R$ die gleichen Kosten. Also weisen L und $\text{REP}_\mu^\pi[k](L, G)$ dieselbe Menge an $\mathbb{SB}_\mu^\pi[k]$ -Partnern auf. Folglich haben L und $\text{REP}_\mu^\pi[k](L, G)$ dieselbe Partnermenge bezüglich $\mathbb{SB}_\mu^\pi[k]$. Laut Lemma 231, S. 175 ist $\text{REP}_\mu^\pi[k](L, G)[\sigma] \in \text{REPS}_\sigma(\text{REP}_\mu^\pi[k](L, G))$. Da L und $\text{REP}_\mu^\pi[k](L, G)$ dieselbe Partnermenge bezüglich $\mathbb{SB}_\mu^\pi[k]$ aufweisen, gilt damit $\text{REP}_\mu^\pi[k](L, G)[\sigma] \in \text{REPS}_\sigma(L)$.

Wir führen die folgenden Notationen für diese kanonischen Partner-Repräsentanten ein:

Notation 233 (Kanonischer Partner-Repräsentant)

Sei $\pi = \text{FILTER}[T']$ eine Filter-Perspektive. Sei G ein k -Graph von $\mathbb{K}[L, \pi]$ in μ . Wir führen die folgenden Notationen ein:

- (1) Wir schreiben $\text{REP}_\sigma(L, G)$ für $\text{REP}_\mu^\pi[k](L, G)[\sigma]$.
- (2) Sei μ assoziativ. Dann schreiben wir $\text{REP}_{\text{ASSOC}_\sigma}(L)$ für das Netz $\text{REP}_{\text{ASSOC}_\sigma}^\pi[k](L)[\sigma]$.

In Abschnitt 9.1.1, S. 167 betrachteten wir die Perspektiven-Repräsentanten aus Abschnitt 4.1, S. 101 bezüglich der Kompositionalität. Daraus schließen wir, dass wir die Aussage von Satz 232 auf einige weitere Perspektiven ausweiten können, wie in Tabelle 2 beschrieben.

Perspektive	Abschnitt	Konstruktion/Einschränkungen
Identität	9.1.1.1	direkt
Anwendungsfall	9.1.1.5	direkt

Tabelle 2 Anwendung der Ergebnisse aus Abschnitt 9.1.1, S. 167

9.2.3 Partner-Repräsentanten für Kosten-Optimalität

Sei L ein Netz über \mathbb{K} . Sei π eine Perspektive von L . Sei μ ein Verbands-Kostenmodell über \mathbb{K} . Sei ρ eine Anforderung. Sei $\succsim = \succsim_\mu^\pi$.

Wenn L nicht $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar ist, können wir aus Gleichung (89), S. 156 folgern, dass alle ρ -Partner optimal sind. In diesem Fall ist L ein Partner-Repräsentant.

Lemma 234

Sei $\sigma = \{\rho \mapsto \rho, \succsim \mapsto \text{INDIFF}\}$. Sei L nicht $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar.

L ist ein σ -Partner-Repräsentant.

Beweis (Lemma 234)

Folgt aus Gleichung (89), S. 156 und $\rho[L]^\succsim \subseteq \rho[L]$.

Wenn L dagegen $\rho \cap \mathbb{B}_\mu^\pi$ -bedienbar ist, ist dies allgemein nicht mehr gegeben. Wenn \leq_μ jedoch eine Wohlordnung ist, können wir die kanonische Schranke $\text{BOUND}_\mu^\pi(L, \rho)$ (siehe Notation 213, S. 160) nutzen.

Lemma 235

Sei \leq_μ eine Wohlordnung bis auf ω . Sei $k = \text{BOUND}_\mu^\pi(L, \rho)$ und $\sigma = \{\rho \mapsto \rho \cap \mathbb{B}_\mu^\pi[k], \succsim \mapsto \text{INDIFF}\}$. Sei L bedienbar bezüglich $\rho \cap \mathbb{B}_\mu^\pi$.

L ist ein σ -Partner-Repräsentant.

Beweis (Lemma 235)

Sei R ein ρ -Partner von L . Aus Notation 206, S. 157 können wir leicht folgern: R ist ein \succsim -optimaler ρ -Partner von L g.d.w. $\llbracket L \oplus R \rrbracket_\mu^\pi = k$. Also gilt $\rho[L]^\succsim = \rho \cap \mathbb{B}_\mu^\pi[k][L]$.

9.3 Diskussion

In diesem Kapitel haben wir zwei Klassen von Repräsentanten vorgestellt:

- In Abschnitt 9.1.2, S. 170 haben wir durch das Konzept der Kompositionalität die Perspektiven-Repräsentanten, Wert-Repräsentanten und Schranken-Repräsentanten geschlossener Systeme aus Kapitel 4, S. 101 von einzelnen Systemen L auf beliebige Systeme $L \oplus R$ übertragen.
- In Abschnitt 9.2, S. 173 haben wir Partner-Repräsentanten eingeführt, die speziell für Analyse und Synthese offener Systeme konzipiert sind.

In Abschnitt 9.1.2, S. 170 haben wir dabei die Ergebnisse aus Abschnitt 3.5, S. 70 mit den Ergebnissen aus Kapitel 4, S. 101 kombiniert. Der Kern bestand darin zu zeigen, dass die Konstruktionen aus Abschnitt 3.5, S. 70 bisimilare Systeme erzeugen. In Abschnitt 9.2, S. 173 haben wir die Ergebnisse aus Abschnitt 9.1.2, S. 170 verwendet, um aus Schranken-Repräsentanten Partner-Repräsentanten zu konstruieren. Dazu haben wir ausgenutzt, dass in Schranken-Repräsentanten die Verletzung der Schranke k durch eine Menge von Markierungen charakterisiert wird. Durch Restriktion der Rahmenmarkierungen auf genau solche, in denen die Kosten k nicht überschritten haben, entstand aus einem Schranken-Repräsentant ein Partner-Repräsentant. Ähnlich hätten wir auch mit Wert-Repräsentanten verfahren können: Ein Wert-Repräsentant ist ebenso kosteneindeutig. Daher hätten wir die Rahmenmarkierungen auch hier entsprechend anpassen können. Da unsere Ergebnisse zur Konstruktion von Schranken-Repräsentanten umfassender als die zur Konstruktion von Wert-Repräsentanten sind, haben wir uns entschieden diese Konstruktion auf Schranken-Repräsentanten zu beschränken und die Konstruktion aus Wert-Repräsentanten nicht zu formalisieren.

Auf den ersten Blick ähnelt das Konzept der Partner-Repräsentanten dem der Austauschbarkeit [Aal+08] aus Definition 185, S. 143. Austauschbarkeit fordert eine Beziehung zwischen den jeweiligen Partnermengen zweier offener Systeme bezüglich einer Anforderung. Wir betrachten dagegen die zwei jeweiligen Partnermengen eines offenen Systems bezüglich zwei Kombinationen aus Anforderung und Präferenz. Daher sind diese Konzepte orthogonal. Erst durch den Begriff eines Partner-Repräsentanten L' eines offenen Systems L wird eine Verbindung hergestellt: L und L' ersetzen sich bezüglich einer Kombination aus Anforderung und Präferenz gegenseitig. Wir zeigen diese Beziehung zwischen L und L' , indem wir die Bisimilarität der Systeme $L \oplus R$ und $L' \oplus R$ für beliebige R zeigen. Auf diese Weise zeigen wir, dass L und R die gleichen \S -Partner haben. Stattdessen hätten wir hier auch einen anderen Ansatz wählen können [SMB09; Moo+11], um die Austauschbarkeit von L und L' zu zeigen: Bisimilarität ist hinreichend jedoch nicht notwendig für Austauschbarkeit.

10 Analyse und Synthese auf Basis von Repräsentanten

In diesem Kapitel beschreiben wir den Nutzen von Repräsentanten und beantworten die in Kapitel 8, S. 165 gestellten Fragen der Bedienbarkeit, der Partnersynthese und der Repräsentation für folgende Anforderungen und Präferenzen: Wir betrachten wir die Anforderungen \mathbb{B}_μ^π , $\mathbb{B}_\mu^\pi[k]$ und $\mathbb{SB}_\mu^\pi[k]$ und die Präferenz INDIFF . Erst in Kapitel 12, S. 195 betrachten wir die Anforderung \mathbb{SB}_μ^π sowie die Präferenz \succ_μ^π .

Wir konzentrieren uns in Abschnitt 10.1 auf das Problem der Bedienbarkeit. In Abschnitt 10.2, S. 185 erklären wir den Nutzen der gewonnenen Ergebnisse für die Lösung der Partnersynthese und der Repräsentation. Die Ergebnisse dieses Kapitels besprechen wir in Abschnitt 10.3, S. 187.

Seien dazu im gesamten Kapitel L , π , μ , k und σ wie folgt definiert: Sei L ein Netz über \mathbb{K} . Sei $\pi = \text{FILTER}[T]$ eine Filter-Perspektive von L und μ ein assoziatives Verbands-Kostenmodell über \mathbb{K} . Sei $k \in \mathbb{K}$ eine Schwelle und $\sigma = \{\mathbb{SB}_\mu^\pi[k] \mapsto \mathbb{S}\}$.

10.1 Entscheidung der Bedienbarkeit bezüglich Kostenbeschränktheit

In diesem Abschnitt betrachten wir die Entscheidbarkeit unterschiedlicher Bedienbarkeitsprobleme. Wir beginnen mit $\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit, betrachten anschließend \mathbb{B}_μ^π -Bedienbarkeit und schließen mit $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit ab.

10.1.1 $\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit

Wir zeigen, dass $\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit mit k -Kostenbeschränktheit von L zusammenfällt:

Lemma 236

$\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit ist k -Kostenbeschränktheit.

Beweis (Lemma 236)

- (1) Wenn L k -kostenbeschränkt ist, können wir ein leeres R angeben, sodass $\text{RUNS}(L \oplus R) = \text{RUNS}(L)$ und somit $\llbracket L \rrbracket_\mu^\pi = \llbracket L \oplus R \rrbracket_\mu^\pi$ gilt. Daher ist ein k -kostenbeschränktes Netz L auch $\mathbb{B}_\mu^\pi[k]$ -bedienbar.
- (2) Wenn L nicht k -kostenbeschränkt ist, gilt für kein R : $\llbracket L \oplus R \rrbracket_\mu^\pi \leq_\mu k$, da $\text{RUNS}(L) \subseteq \text{RUNS}(L \oplus R)$ gilt. Also ist ein nicht k -kostenbeschränktes Netz nicht $\mathbb{B}_\mu^\pi[k]$ -bedienbar.

Durch Anwendung von Satz 166, S. 122 aus Abschnitt 5.2, S. 120 erhalten wir direkt:

Korollar 237

Sei $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ endlich. Dann ist die $\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit von L entscheidbar.

Lemma 236, S. 181 gilt auch dann, wenn wir π durch eine der folgenden Perspektiven ersetzen:

- (1) ID oder
- (2) $\text{AF}[L, U]$ für einen Anwendungsfall U

In beiden Fällen nutzen wir, dass das Verhalten von L bezüglich der Perspektiven jeweils ein Teilverhalten von $L \oplus R$ bezüglich der entsprechenden Perspektiven ist.

Aus Korollar 167, S. 123 können wir dann direkt folgern:

Korollar 238

Sei U ein Anwendungsfall von L . Sei

$$\pi' \in \{\text{ID}, \text{AF}[L, U]\} .$$

Sei $(\mathbb{K}[L, \pi'])_{\leq k}^\mu$ endlich.

Dann ist $\mathbb{B}_\mu^{\pi'}[k]$ -Bedienbarkeit bezüglich π' von L entscheidbar.

Bei der Terminierungs-Perspektive sieht es anders aus. Hier können wir leicht folgern, dass L k -kostenbeschränkt bedienbar ist: Sei R ein beliebiges Netz mit einer leeren Menge von Zielmarkierungen. Dann ist die Menge der Zielmarkierungen von $L \oplus R$ ebenfalls leer. Somit ist auch die Menge der terminierenden Abläufe von $L \oplus R$ leer. Daraus schließen wir $\llbracket L \oplus R \rrbracket_\mu^\pi \leq_\mu k$. Folglich gilt:

Korollar 239

Sei $\pi = \text{TERM}[L]$. Dann ist L $\mathbb{B}_\mu^\pi[k]$ -bedienbar.

10.1.2 \mathbb{B}_μ^π -Bedienbarkeit

Das Problem der \mathbb{B}_μ^π -Bedienbarkeit ähnelt der im vorherigen Abschnitt 10.1.1, S. 181 behandelten $\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit. Bei der \mathbb{B}_μ^π -Bedienbarkeit wird jedoch kein festes k angenommen, sondern die Existenz eines k ungleich des Supremums \top von \mathbb{K} gefordert.

Dennoch haben die beiden Probleme eine Gemeinsamkeit: Nach Lemma 236, S. 181 koinzidieren $\mathbb{B}_\mu^\pi[k]$ -Bedienbarkeit und k -Kostenbeschränktheit. Hier können wir die analoge Koinzidenz zeigen:

Korollar 240

\mathbb{B}_μ^π -Bedienbarkeit ist Kostenbeschränktheit.

Wir können also die Resultate aus Abschnitt 5.3, S. 123 anwenden, um \mathbb{B}_μ^π -Bedienbarkeit zu entscheiden.

Durch Anwendung von Korollar 169, S. 125 aus Abschnitt 5.3, S. 123 erhalten wir direkt:

Korollar 241

Sei $T' \subseteq T$ eine Menge von Transitionen von L , U ein Anwendungsfall von L . Sei

$$\pi' \in \{\text{FILTER}[T], \text{ID}, \text{AF}[L, U]\}.$$

Sei μ durch die Max-Plus-Algebra induziert oder $(\mathbb{K}[L, \pi'])^\mu$ endlich.

Dann ist die \mathbb{B}_μ^π -Bedienbarkeit bezüglich π' von L entscheidbar.

Analog zu Korollar 239, S. 182 können wir beobachten, dass aus der Terminierungs-Perspektive L stets kostenbeschränkt bedienbar ist.

Korollar 242

Sei $\pi = \text{TERM}[L]$. Dann ist L \mathbb{B}_μ^π -bedienbar.

10.1.3 $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit

Wir untersuchen abschließend die Entscheidbarkeit der $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit. Dafür reduzieren wir $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit auf \mathbb{S} -Bedienbarkeit indem wir die Ergebnisse aus Abschnitt 9.1.3, S. 173 und Abschnitt 9.2.2, S. 175 anwenden. Die Abbildung 45 veranschaulicht unseren Ansatz: Zu einem offenen Netz L konstruieren wir einen äquivalenten Partner-Repräsentanten L' . Anschließend entscheiden wir \mathbb{S} -Bedienbarkeit von L' und übertragen das Ergebnis auf die $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L .

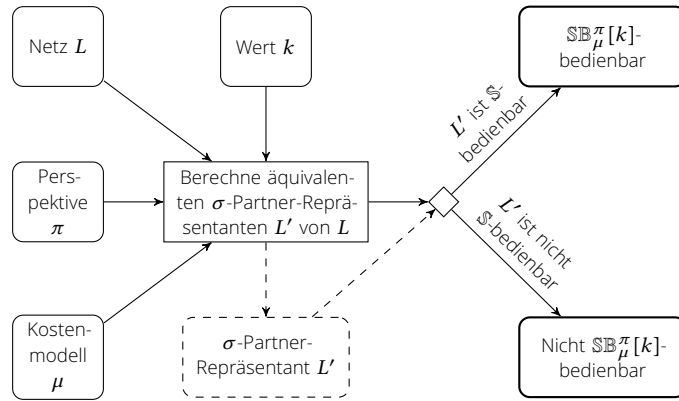


Abbildung 45 Unser Ansatz zur Entscheidung der Bedienbarkeit bezüglich schwacher Terminierung und k -Kostenbeschränktheit

Für einen σ -Partner-Repräsentanten L ist die Reduktion von $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit auf \mathbb{S} -Bedienbarkeit trivial, denn es gilt: $\mathbb{SB}_\mu^\pi[k][L] = \mathbb{S}[L]$.

Korollar 243

Sei L ein σ -Partner-Repräsentant. Dann ist die $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L entscheidbar.

Wenn L kein σ -Partner-Repräsentant ist, ist die Reduktion nicht so trivial. Für unseren Ansatz überführen wir L in einen σ -Partner-Repräsentanten.

Lemma 244

Existiere ein Algorithmus, der aus L einen σ -Partner-Repräsentanten $L' \in \text{REPS}_\sigma(L)$ berechnet. Dann ist $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L entscheidbar.

Beweis (Lemma 244)

Aus $L' \in \text{REPS}_\sigma(L)$ folgt die Koinzidenz der $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L und L' . Wir können die $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L entscheiden, indem wir zuerst L' berechnen und anschließend $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L' entscheiden (siehe Korollar 243).

Wir betrachten daher folgend die Fälle, in denen wir einen Partner-Repräsentanten $L' \in \text{REPS}_\sigma(L)$ aus L berechnen können. Laut Satz 232, S. 177 ist $\text{REPAssoc}_\sigma(L)$ (siehe Notation 233, S. 177) ein Partner-Repräsentant von L . Wenn der k -Abschluss von N in μ endlich ist, können wir $\text{REPAssoc}_\sigma(L)$ wie folgt berechnen:

- (1) Berechne den kanonischen k -Graphen G von $\mathbb{K}[L, \pi]$ in μ . Dieser ist laut Korollar 76, S. 59 endlich.

- (2) Berechne $\text{REPAssoc}_\mu^\pi[k](L)$ aus L und G . Die Berechnung findet direkt auf der Struktur statt und das Resultat ist laut Korollar 129, S. 86 endlich.
- (3) Berechne $\text{REPAssoc}_\sigma(L)$ aus $\text{REPAssoc}_\mu^\pi[k](L)$ durch Anpassung der Rahmenmarkierungen. Ist die Menge der Rahmenmarkierungen vorher als ein Mengenausdruck E kodiert, dann können wir die neue Menge der Rahmenmarkierungen leicht als den folgenden Mengenausdruck kodieren:

$$\{m \mid m \text{ matcht } e, m(\frac{e}{\downarrow}) = 0\} \text{ .}$$

Daraus folgern wir:

Korollar 245

Sei $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ endlich. Dann ist die $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit von L entscheidbar.

Wie schon im vorherigen Abschnitt 10.1.1, S. 181 weiten wir auch hier die Ergebnisse auf weitere Perspektiven aus. Auf Basis von Tabelle 2, S. 178 folgern wir:

Korollar 246

Sei U ein Anwendungsfall von L . Sei

$$\pi' \in \{\text{ID}, \text{AF}[L, U]\} \text{ .}$$

Sei $(\mathbb{K}[L, \pi'])_{\leq k}^\mu$ endlich.

Dann ist $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit bezüglich π' von L entscheidbar.

In Kapitel 12, S. 195 betrachten wir die Entscheidbarkeit der \mathbb{SB}_μ^π -Bedienbarkeit durch eine Reduktion auf $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit.

10.2 Partnersynthese und Repräsentation für Kostenbeschränktheit

Wir untersuchen im Folgenden die Übertragbarkeit der Ergebnisse aus Abschnitt 10.1, S. 181 auf die Probleme der Partnersynthese und der Repräsentation. Dabei betrachten wir in dieser Reihenfolge $\mathbb{B}_\mu^\pi[k]$, \mathbb{B}_μ^π und $\mathbb{SB}_\mu^\pi[k]$.

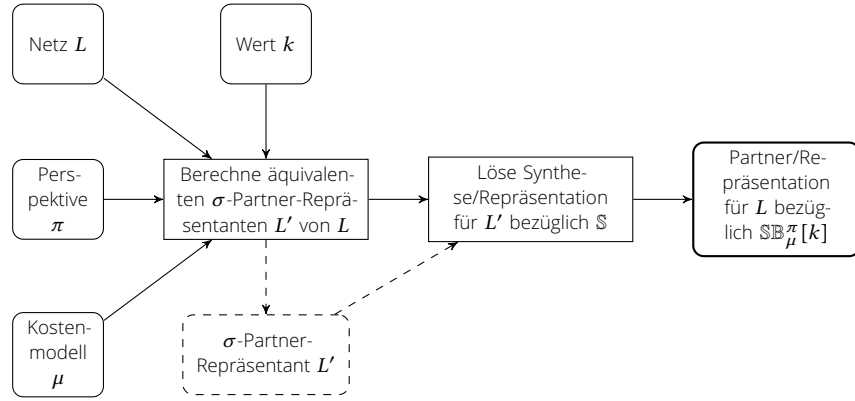


Abbildung 46 Unser Ansatz für Partnersynthese und Repräsentation bezüglich schwacher Terminierung und k-Kostenbeschränktheit

10.2.1 Partnersynthese und Repräsentation für $\mathbb{B}_\mu^\pi[k]$

Sei $L \mathbb{B}_\mu^\pi[k]$ -bedienbar. Sei R ein offenes System, sodass die Komposition von L und R ein geschlossenes System ist, und R keine Transition enthält. Dann ist R ein $\mathbb{B}_\mu^\pi[k]$ -Partner. Folglich ist Partnersynthese für $\mathbb{B}_\mu^\pi[k]$ trivial lösbar. Die Repräsentation ist dagegen schwieriger und wird in dieser Arbeit nicht betrachtet.

10.2.2 Partnersynthese und Repräsentation für \mathbb{B}_μ^π

Sei $L \mathbb{B}_\mu^\pi$ -bedienbar. Analog zum vorherigen Abschnitt 10.2.1, argumentieren wir auch hier, dass Partnersynthese trivial durch Angabe eines kanonischen, leeren offenen Systems R lösbar ist: Wenn R keine Transitionen enthält und $L \oplus R$ geschlossen ist, ist R ein \mathbb{B}_μ^π -Partner von L . Die Repräsentation ist dagegen schwieriger und wird in dieser Arbeit nicht betrachtet.

10.2.3 Partnersynthese und Repräsentation für $\mathbb{SB}_\mu^\pi[k]$

In Abschnitt 10.1.3, S. 183 beschreiben wir den Nutzen von Partner-Repräsentanten zur Entscheidung der $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit. Dabei nutzen wir aus, dass die Menge der $\mathbb{SB}_\mu^\pi[k]$ -Partner und die Menge der \mathbb{S} -Partner eines σ -Partner-Repräsentanten gleich sind. Wir argumentieren analog, um Partnersynthese und Repräsentation zu lösen: Ist L ein σ -Partner-Repräsentant, können wir $\mathbb{SB}_\mu^\pi[k]$ -Partnersynthese ($\mathbb{SB}_\mu^\pi[k]$ -Repräsentation) auf \mathbb{S} -Partnersynthese (\mathbb{S} -Repräsentation) reduzieren. Die Abbildung 46 veranschaulicht diesen Ansatz.

Können wir einen zu L äquivalenten σ -Partner-Repräsentanten L' berechnen, dann können wir die Reduktion analog durchführen: Wir reduzieren dazu $\mathbb{SB}_\mu^\pi[k]$ -Partnersynthese ($\mathbb{SB}_\mu^\pi[k]$ -Repräsentation) für L auf \mathbb{S} -Partnersynthese (\mathbb{S} -Repräsentation) für L' . Korollar 245, S. 185 und Korollar 246, S. 185 beschreiben Fälle, für die wir L' für L bestimmen können. Daraus schließen wir die Lösbarkeit der Partnersynthese für Filter-, Identitäts- und Anwendungsfall-Perspektiven:

Korollar 247

Sei $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ endlich.

Dann sind Partnersynthese und Repräsentation für L bezüglich $\mathbb{SB}_\mu^\pi[k]$ lösbar.

Korollar 248

Sei $(\mathbb{K}[L, \pi'])_{\leq k}^\mu$ endlich. Sei U ein Anwendungsfall von L . Sei

$$\pi' \in \{\text{ID}, \text{AF}[L, U]\} \quad .$$

Dann sind Partnersynthese und Repräsentation für L bezüglich $\mathbb{SB}_\mu^\pi[k]$ bezüglich π' lösbar.

10.3 Diskussion

In diesem Kapitel haben wir Lösungen für die in Kapitel 8, S. 165 auf Basis von Repräsentanten angegeben. Dabei haben wir stets ein Problem mit einem nicht-funktionalen Aspekt, wie beispielsweise Kostenbeschränktheit, auf ein Problem ohne nicht-funktionale Aspekte reduziert. Die einzige Ausnahme bildet auf den ersten Blick die Entscheidung der Bedienbarkeit für Kostenbeschränktheit bzw. k -Kostenbeschränktheit, die wir auf Kostenbeschränktheit bzw. k -Kostenbeschränktheit reduziert haben. Lösen wir das Problem der Kostenbeschränktheit bzw. k -Kostenbeschränktheit jedoch mit den Methoden aus Kapitel 5, S. 117, findet auch hier eine Reduktion auf ein Problem ohne nicht-funktionale Aspekte statt. Wir profitieren hierbei von der Existenz von Methoden für die Analyse und Synthese offener Systeme bezüglich der funktionalen Verhaltens-Anforderung der schwachen Terminierung wie zum Beispiel [Wolog]. Da wir vorwiegend Reduktionen nicht-funktionaler Aspekte auf funktionale Aspekte vornehmen, sind wir nicht an einen konkreten Ansatz zur Lösung der Probleme für funktionale Verhaltens-Anforderungen gebunden.

Die betrachteten Probleme der Bedienbarkeit, Partnersynthese und Repräsentation sind altbekannte Probleme aus dem Bereich der temporalen Logik.

Für eine Übersicht über die bestehende Techniken zur Partnersynthese empfehlen wir [Hei14]. Die bekannten Arbeiten aus diesem Bereich wie [PR90; KM00; AV03; Blo+09; Ben+10] unterscheiden sich durch das gewählte Kommunikationsmodell und der Klasse der betrachteten Verhaltens-Anforderungen. Im Gegensatz zu den vorliegenden Arbeiten betrachten wir als Kommunikationsmodell gepufferte, ungeordnete, asynchrone Kommunikationsmodell. Außerdem betrachten wir Kosten und damit auch nicht-funktionale Aspekte. Dafür betrachten die vorliegenden Arbeiten größere Klassen von funktionalen Verhaltens-Anforderungen: Häufig wird eine beliebige temporallogische Formel aus einer gegebenen Logik angenommen. Wie oben beschrieben sind wir frei von dem tatsächlichen Ansatz zur Lösung der Analyse- und Synthese-Probleme. Die Anwendbarkeit unserer Ansätze in Bezug auf andere Klassen von Verhaltens-Anforderungen diskutieren wir auch in Abschnitt 14.2, S. 220.

11 Kostendiskriminanten offener Systeme

In diesem Kapitel betrachten wir *Kostendiskriminanten*, die wir als Grundlage zur Entscheidung der kostenbeschränkten Bedienbarkeit verwenden. Sei dazu L ein gewichtetes Netz über \mathbb{K} , π eine Perspektive von L und $\langle \mathbb{K}, \leq \rangle$ ein vollständiger Verband mit größtem Element \top . Sei ρ eine Anforderung. Ein Wert $\delta \in \mathbb{K}$ ist eine *Kostendiskriminante* von L bezüglich π , μ , \leq und ρ , falls $\rho \cap \mathbb{B}_{\mu, \leq}^{\pi}[\delta]$ -Bedienbarkeit und $\rho \cap \mathbb{B}_{\mu, \leq}^{\pi}$ -Bedienbarkeit zusammenfallen.

Kostendiskriminanten

In Abschnitt 11.1 führen wir formal Kostendiskriminanten für beliebige Kostenmodelle ein. In Abschnitt 11.2 betrachten wir Kostendiskriminanten für Verbands-Kostenmodelle (siehe Abschnitt 11.2.1) und die Max-Plus-Algebra (siehe Abschnitt 11.2.2, S. 193). In Abschnitt 11.3, S. 194 diskutieren wir die Ergebnisse dieses Kapitels.

11.1 Kostendiskriminanten für beliebige Kostenmodelle

Wir formalisieren den Begriff der Kostendiskriminante.

Definition 249 (Kostendiskriminante)

Sei L ein gewichtetes Netz über \mathbb{K} , π eine Perspektive von L und $\langle \mathbb{K}, \leq \rangle$ ein vollständiger Verband mit größtem Element \top . Sei ρ eine Anforderung.

Ein $\delta \neq \top \in \mathbb{K}$ ist eine *Kostendiskriminante* von L bezüglich π , μ , \leq und ρ , falls die folgende Aussage erfüllt ist:

$$(\mathbb{B}_{\mu, \leq}^{\pi} \cap \rho)[L] \neq \emptyset \text{ g.d.w. } (\mathbb{B}_{\mu, \leq}^{\pi}[\delta] \cap \rho)[L] \neq \emptyset . \quad (98)$$

Möchten wir die Gültigkeit der Aussage (98) für gegebene L , π , μ , \leq und ρ zeigen, ist es notwendig und hinreichend die Gültigkeit der folgenden Aussage zu zeigen:

$$(\mathbb{B}_{\mu, \leq}^{\pi} \cap \rho)[L] \neq \emptyset \text{ impliziert } (\mathbb{B}_{\mu, \leq}^{\pi}[\delta] \cap \rho)[L] \neq \emptyset . \quad (99)$$

Aussage (99) beschreibt nur die eine Richtung der Aussage (98). Die verbleibende Richtung folgt daraus, dass jedes k -kostenbeschränkt bedienbare Netz L auch kostenbeschränkt bedienbar ist.

11.2 Konstruktion von Kostendiskriminanten

In Abschnitt 11.2.1 konstruieren wir eine kanonische Kostendiskriminante für eine Teilklasse der Verbands-Kostenmodelle. In Abschnitt 11.2.2, S. 193 zeigen wir, dass das von der Max-Plus-Algebra induzierte Kostenmodell ein Element dieser Teilklasse ist.

11.2.1 Kostendiskriminanten für Verbands-Kostenmodelle

Wir betrachten ein gewichtetes Netz L über \mathbb{K} , eine Filter-Perspektive $\pi = \text{FILTER}[T]$ sowie ein assoziatives Verbands-Kostenmodell μ über \mathbb{K} , wobei \top das größte Element bezüglich \leq_μ ist. Sei das Supremum jeder endlichen Teilmenge von \mathbb{K} ungleich \top .

Wir betrachten im Folgenden besondere Umstände, unter denen eine Kostendiskriminante für die Anforderung der schwachen Terminierung aus der Komposition $L \oplus \text{MPP}(L)$ aus L und dem most-permissive \mathbb{S} -Partner von L konstruiert werden kann. Genauer zeigen wir unter welchen Annahmen gilt, dass das Supremum der Kosten der zyklensfreien Abläufe in $L \oplus \text{MPP}(L)$ eine Kostendiskriminante ist. Wir argumentieren dabei folgendermaßen:

- (1) Unbeschränkte Kosten entstehen immer durch Zyklen, die Kosten verursachen. Solche Zyklen nennen wir *kostensteigernd*.
- (2) Jeder Zyklus in einem Kompositum $L \oplus R$ entspricht einem L -Zyklus (siehe Abschnitt 7.2.2, S. 146).
- (3) Kann wenigstens ein \mathbb{S} -Partner einen L -Zyklus beschränkt überdecken, kann der Zyklus auch verhindert werden (siehe Lemma 194, S. 151).
- (4) Ist L kostenbeschränkt bedienbar, kann also jeder der kostensteigernde Zyklen vermieden werden.
- (5) Der most-permissive \mathbb{S} -Partner überdeckt jeden überdeckbaren L -Zyklus (folgt aus der Definition von L -Zyklus).
- (6) Daher betrachten wir die zyklensfreie Abläufe in $L \oplus \text{MPP}(L)$.

Wir nennen einen L -Zyklus z *kostensteigernd*, falls für wenigstens einen Ablauf $w = uv$ in $L \oplus \text{MPP}(L)$ gilt: w überdeckt z mit v und $\mu(u|_T) < \mu(w|_T)$. Das heißt, Ausführen des Zyklus steigert die Kosten. Sei folgend $Z_>$ die Menge der kostensteigernden L -Zyklen. Die Frage nach der Zugehörigkeit von z zu $Z_>$ ist nicht trivial: Im Allgemeinen können unendlich viele Abläufe z überdecken.

Wir betrachten im Folgenden die Situation, dass die Menge $\mathbb{K}[L, \pi]$ drei wichtige Eigenschaften aufweist:

- (1) Der Abschluss von $\mathbb{K}[L, \pi]$ enthält nicht das Top-Element \top . Dann können entlang eines Ablaufs stets nur beschränkte Kosten entstehen.
- (2) Die Menge $\mathbb{K}[L, \pi]$ ist monoton. Das heißt, mehrmalige Ausführung eines kostensteigernden Zyklus führt stets zu höheren Kosten.
- (3) Der k -Abschluss von $\mathbb{K}[L, \pi]$ ist für alle $k \neq \top$ endlich, das heißt, steigen Kosten immer weiter, wird jede Schranke k irgendwann überschritten.

Ist das Kompositum $L \oplus \text{MPP}(L)$ beschränkt, induzieren die zyklensfreien Abläufe von $L \oplus \text{MPP}(L)$ eine Kostendiskriminante. Dies zeigen wir in mehreren Schritten.

Sei R ein \mathbb{S} -Partner von L . In Abschnitt 7.2.2, S. 146 haben wir die Notation $R[z]$ für die Anzahl der Überdeckungen eines L -Zyklus z durch R eingeführt. Gilt $R[z] = 0$ für alle $z \in Z_{>}$, dann überdeckt auch $L \oplus R$ keinen kostensteigernden Zyklus. Aus den ersten beiden Eigenschaften folgern wir die Kostenbeschränktheit von $L \oplus R$:

Lemma 250

Sei $L \oplus \text{MPP}(L)$ beschränkt. Gelte (1) $\top \notin (\mathbb{K}[L, \pi])^\mu$ und sei (2) $\mathbb{K}[L, \pi]$ monoton in μ . Sei R ein \mathbb{S} -Partner von L mit $R[z] = 0$ für alle $z \in Z_{>}$.

Dann ist $L \oplus R$ kostenbeschränkt.

Beweis (Lemma 250)

Wir zeigen, dass die Kosten von $L \oplus R$ einzig von den zyklensfreien Abläufen sind. Seien u, v, w Transitionssequenzen, so dass für $uv^n w$ ein Ablauf in $L \oplus R$ für alle $n \in \mathbb{N}_0$ ist. Da $L \oplus R$ beschränkt ist, überdecken für ein $k > 0$ alle Abläufe $uv^\ell w$ mit $\ell \geq k$ einen L -Zyklus z wenigstens ℓ mal. Also überdeckt R einen L -Zyklus z unbeschränkt. Per Annahme gilt dann $z \notin Z_{>}$. Folglich gilt $\mu(uv^n) = \mu(u)$. Aus (2) und der Assoziativität von μ können wir dann $\mu(uv^n w) = \mu(uw)$ folgern. Also existiert zu jedem beliebigen Ablauf ein entsprechender kostenäquivalenter zyklensfreier Ablauf. Aus (1) folgern wir $\llbracket L \oplus R \rrbracket_\mu^\pi \neq \top$.

Durch Hinzunahme der Eigenschaft (3) können wir zeigen, dass L genau dann \mathbb{SB}_μ^π -bedienbar ist, wenn ein \mathbb{S} -Partner existiert, der keinen kostensteigernden Zyklus überdeckt.

Lemma 251

Sei $L \oplus \text{MPP}(L)$ beschränkt. Gelte (1) $\top \notin (\mathbb{K}[L, \pi])^\mu$ und sei (2) $\mathbb{K}[L, \pi]$ monoton in μ . Gelte (3) für alle $k \neq \top \in \mathbb{K}$: $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ ist endlich.

Dann sind die beiden folgenden Aussagen äquivalent:

- (1) L ist ein \mathbb{SB}_μ^π -bedienbares Netz.
- (2) Es existiert ein \mathbb{S} -Partner R mit $f_R(z) = 0$ für alle $z \in Z_{>}$.

Beweis (Lemma 251)

- (1) Sei L \mathbb{SB}_μ^π -bedienbar. Dann existiert ein \mathbb{SB}_μ^π -Partner R von L . Sei $\llbracket L \oplus R \rrbracket_\mu^\pi = k$. Dann gilt für alle Abläufe w von $L \oplus R$: w hat höchstens Kosten k . Also überdeckt w jeden Zyklus $z \in Z_{>}$ nur beschränkt häufig: Per (3) ist der k -Abschluss endlich und per (2) ist die Menge $\mathbb{K}[L, \pi]$ monoton in μ . Also gilt $f_R(z) \leq_\mu m$ für eine natürliche Zahl m . Aus Lemma 194, S. 151 folgt die Existenz eines \mathbb{S} -Partners R' mit $f_{R'}(z) = 0$ für alle $z \in Z_{>}$.
- (2) Sei R ein \mathbb{S} -Partner mit $f_R(z) = 0$ für alle $z \in Z_{>}$. Dann folgt aus Lemma 250, S. 191: $L \oplus R$ ist kostenbeschränkt. Also ist L \mathbb{SB}_μ^π -bedienbar.

Die Kosten eines solchen Partners R aus Satz 252 hängen also eng mit den zyklensfreien Abläufen von $L \oplus \text{MPP}(L)$ zusammen. Tatsächlich induziert die Menge der zyklensfreien Abläufe von $L \oplus \text{MPP}(L)$ eine Kostendiskriminante:

Satz 252 (Kostendiskriminante)

Sei $L \oplus \text{MPP}(L)$ beschränkt. Gelte (1) $\top \notin (\mathbb{K}[L, \pi])^\mu$ und sei (2) $\mathbb{K}[L, \pi]$ monoton in μ . Gelte (3) für alle $k \neq \top \in \mathbb{K}$: $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ ist endlich. Sei $\pi' = \text{ZF}[L \oplus \text{MPP}(L)] \triangleright \pi$. Sei $\delta = \llbracket L \oplus \text{MPP}(L) \rrbracket_\mu^{\pi'}$.

Dann ist δ eine Kostendiskriminante von L bezüglich π , μ , \leq_μ und \mathbb{S} .

Beweis (Satz 252, S. 192)

Wir zeigen die Aussage (99):

$$\mathbb{B}_\mu^\pi \cap \mathbb{S}[L] \neq \emptyset \text{ impliziert } \mathbb{B}_\mu^\pi[\delta] \cap \mathbb{S}[L] \neq \emptyset .$$

Gelte $\mathbb{B}_\mu^\pi \cap \mathbb{S}[L] \neq \emptyset$. Dann existiert nach Lemma 251, S. 192 ein \mathbb{S} -Partner R mit $f_R(z) = 0$ für alle $z \in Z_>$. Sei w ein Ablauf von $L \oplus R$. Da w keinen L -Zyklus aus $Z_>$ überdeckt, gilt für alle von w überdeckten L -Zyklen z : z ist nicht kostensteigernd. Sei v die Projektion von w auf die Transitionen, die keine Zyklen überdecken. Dann sind w und v kostenäquivalent. Außerdem entspricht v einem zyklensfreien Ablauf in $L \oplus \text{MPP}(L)$. Also hat w die Kosten eines zyklensfreien Ablaufs aus $L \oplus \text{MPP}(L)$. Also betragen die Kosten von w höchstens δ . Also gilt $\llbracket L \oplus R \rrbracket_\mu^\pi \leq_\mu \delta$. Also gilt $R \in \mathbb{B}_\mu^\pi[\delta] \cap \mathbb{S}[L]$ und damit auch $\mathbb{B}_\mu^\pi[\delta] \cap \mathbb{S}[L] \neq \emptyset$.

Wir nennen diese Kostendiskriminante δ die *kanonische Kostendiskriminante* von L :

Definition 253 (Kanonische Kostendiskriminante)

Sei $L \oplus \text{MPP}(L)$ beschränkt. Gelte für alle $k \neq \top \in \mathbb{K}$: (1) $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ ist endlich, und (2) $\top \notin (\mathbb{K}[L, \pi])_{\leq k}^\mu$. Sei $\mathbb{K}[L, \pi]$ monoton in μ . Sei $\pi' = \text{ZF}[L \oplus \text{MPP}(L)] \triangleright \pi$.

$\text{CDIS}(L) := \llbracket L \oplus \text{MPP}(L) \rrbracket_\mu^{\pi'}$ ist die *kanonische Kostendiskriminante* von L .

11.2.2 Kostendiskriminanten für die Max-Plus-Algebra

Wir betrachten das von der *Max-Plus-Algebra* $\text{MP} = \langle \mathbb{N}_0 \cup \{-\infty\}, \max, + \rangle$ induzierte Kostenmodell μ . Sei L ein Netz über \mathbb{N}_0 und π eine Filter-Perspektive. Wir beobachten, dass für jede endliche Menge $X \subseteq \mathbb{N}_0$ gilt:

- (1) X ist monoton.
- (2) Der k -Abschluss von X ist für alle $k \neq \omega$ endlich.

Also können wir den Ansatz aus dem vorherigen Abschnitt 11.2.1, S. 190 verwenden. Dabei lässt sich für einen L -Zyklus z die Zugehörigkeit von z zur Menge $Z_>$ der kostensteigernden L -Zyklen leichter beantworten: Existiert wenigstens eine Überdeckung von z , in der eine Transition $t \in T$ mit Kosten > 0 vorkommt, ist z kostensteigernd.

11.3 Diskussion

In diesem Abschnitt diskutieren wir die Ergebnisse dieses Kapitels.

11.3.1 Monotonie als Voraussetzung

In diesem Kapitel haben wir die Monotonie eines Kostenmodells angenommen, um sicherzustellen, dass das beliebig häufige Durchlaufen eines kostensteigernden Zyklus zu unbeschränkten Kosten führt. Dies ähnelt der als *Monotonie des Schaltens* bekannten Eigenschaft von Petrinetzen: Ist eine Transition t in einer Markierung m aktiviert, ist t auch in jeder Markierung $m' \geq m$ aktiviert. Die *Monotonie des Schaltens* ist das Kernargument in der Entscheidbarkeit der Beschränktheit von Petrinetzen. Wir vermuten daher, dass die Berechnung einer Kostendiskriminante ohne Forderung der Monotonie nicht allgemein möglich ist. Wir haben jedoch keinen Beweis für diese Vermutung.

11.3.2 Kleine Kostendiskriminanten

Der Wert $\text{BOUND}_\mu^\pi(L, \mathbb{S})$ ist stets die kleinste Kostendiskriminante eines SB_μ^π -bedienbaren Netzes. Daher ist es aus theoretischer Sicht eine interessante Fragestellung, ob andere kanonische Kostendiskriminante zwischen $\text{BOUND}_\mu^\pi(L, \mathbb{S})$ und $\text{CDIS}(L)$ liegen. Dazu könnte man im Zustandsraum von $L \oplus \text{MPP}(L)$ bewusst zwischen internen und externen Entscheidungen unterscheiden, das heißt, ob L oder $\text{MPP}(L)$ das weitere Verhalten entscheidet. Im Falle einer externen Entscheidung, der nur noch azyklisches Verhalten folgt, könnte man anstatt eines Wertes δ für jeden Zweig i ein δ_i berechnen. Wir vermuten, dass das kleinste δ_i wieder eine Kostendiskriminante ist.

12 Analyse und Synthese auf Basis von Kostendiskriminanten

In diesem Kapitel verwenden wir *Kostendiskriminanten*, um die folgenden Probleme zu lösen:

- (1) Entscheidung der Bedienbarkeit bezüglich der Anforderung \mathbb{SB}_μ^π (siehe Abschnitt 12.1),
- (2) Partnersynthese für die Anforderung \mathbb{SB}_μ^π (siehe Abschnitt 12.2, S. 197),
- (3) Entscheidung der optimalen Bedienbarkeit bezüglich der Präferenz \succ_μ^π und der Anforderung \mathbb{S} (siehe Abschnitt 12.3, S. 198),
- (4) Partnersynthese und Repräsentation für die Anforderung \mathbb{S} und die Präferenz \succ_μ^π (siehe Abschnitt 12.4, S. 198).

Wir diskutieren unsere Ergebnisse in Abschnitt 12.5, S. 202.

Seien dazu im gesamten Kapitel L , π und μ definiert wie folgt: Sei L ein Netz über \mathbb{K} . Sei $\pi = \text{FILTER}[T]$ eine Filter-Perspektive von L und μ ein assoziatives Verbands-Kostenmodell über \mathbb{K} .

12.1 Entscheidung der \mathbb{SB}_μ^π -Bedienbarkeit

In diesem Abschnitt beschreiben wir einen Ansatz zur Entscheidung der \mathbb{SB}_μ^π -Bedienbarkeit mit Kostendiskriminanten. Die Abbildung 47 veranschaulicht unseren Ansatz: Zunächst berechnen wir die kanonische Kostendiskriminante $\delta = \text{CDIS}(L)$. Anschließend entscheiden wir Bedienbarkeit bezüglich $\mathbb{SB}_\mu^\pi[\delta]$. Da δ eine Kostendiskriminante ist, können wir aus dieser Entscheidung direkt die Entscheidung für \mathbb{SB}_μ^π -Bedienbarkeit folgern.

Sei k eine Kostendiskriminante von L . Dann können wir \mathbb{SB}_μ^π -Bedienbarkeit per Definition 249, S. 189 direkt auf $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit reduzieren.

Im vorherigen Abschnitt 11.2, S. 190 haben wir eine kanonische Kostendiskriminante $\text{CDIS}(L)$ für L definiert (siehe Definition 253, S. 193), falls $L \oplus \text{MPP}(L)$ beschränkt ist und die folgenden Bedingungen erfüllt sind:

- (1) Der Abschluss von $\mathbb{K}[L, \pi]$ enthält nicht das Top-Element \top .
- (2) Die Menge $\mathbb{K}[L, \pi]$ ist monoton.

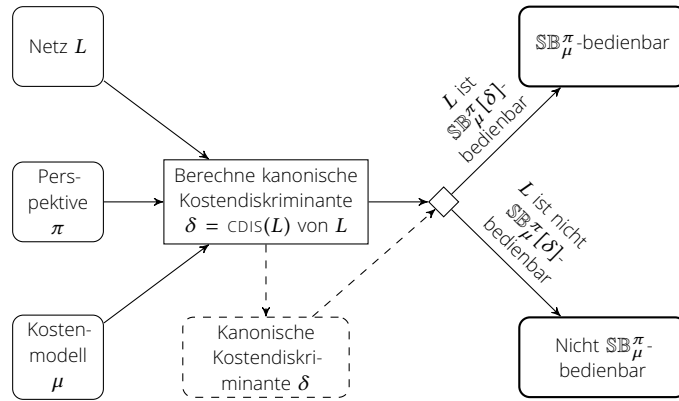


Abbildung 47 Unser Ansatz zur Entscheidung der \mathbb{SB}_μ^π -Bedienbarkeit mit Kostendiskriminanten

(3) Der k -Abschluss von $\mathbb{K}[L, \pi]$ ist für alle $k \neq \top$ endlich.

Die Beschränktheit von $L \oplus \text{MPP}(L)$ ist für alle zu analysierenden Netze L gegeben (siehe Behauptung 189, S. 146 und die Annahme in Kapitel 8, S. 165). Die kanonische Kostendiskriminante $\text{CDIS}(L)$ lässt sich durch Berechnung von $\text{MPP}(L)$ und der anschließenden Betrachtung der endlich vielen zyklensfreien Abläufe von $L \oplus \text{MPP}(L)$ bestimmen. Sind die drei Bedingungen erfüllt, können wir \mathbb{SB}_μ^π -Bedienbarkeit auf $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit reduzieren.

Die Korollare 245 und 246, S. 185 beschreiben Vorbedingungen für die Entscheidbarkeit der $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit. Wir beobachten, dass diese Bedingungen durch die Bedingungen (1) - (3) abgegolten sind. Daraus können wir folgern:

Korollar 254

Sei $L \oplus \text{MPP}(L)$ beschränkt. Gelte für alle $k \neq \omega \in \mathbb{K}$: (1) $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ ist endlich, und (2) $\omega \notin (\mathbb{K}[L, \pi])_{\leq k}^\mu$. Sei $\mathbb{K}[L, \pi]$ monoton in μ .

Dann ist die \mathbb{SB}_μ^π -Bedienbarkeit von L entscheidbar.

Außer für Filter-Perspektive gilt dies auch für die Identität und Anwendungsfall-Perspektiven – in diesem Fall würden wir zunächst einen kompositionalen Perspektiven-Repräsentanten von L konstruieren und anschließend diesen analysieren.

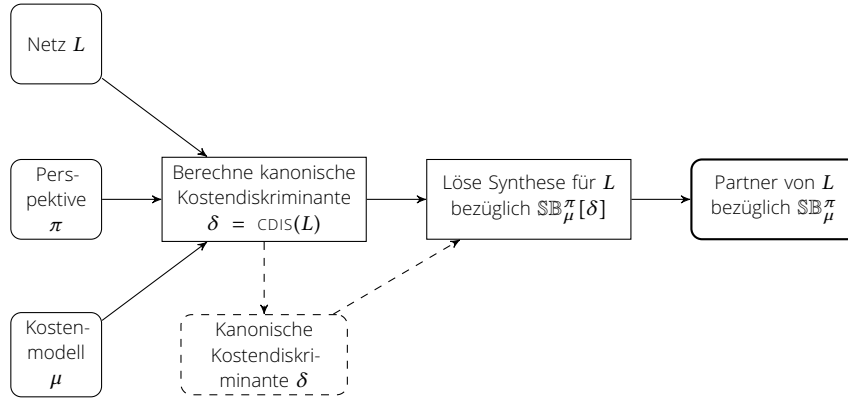


Abbildung 48 Unser Ansatz für Partnersynthese und Repräsentation bezüglich schwacher Terminierung und Kostenbeschränktheit

Korollar 255

Sei U ein Anwendungsfall von L . Sei

$$\pi' \in \{\text{ID}, \text{AF}[L, U]\} \quad .$$

Sei $(\mathbb{K}[L, \pi'])_{\leq k}^\mu$ endlich.

Dann ist die SB_μ^π -Bedienbarkeit bezüglich π' von L entscheidbar.

12.2 Partnersynthese für schwache Terminierung und Kostenbeschränktheit

In diesem Abschnitt schlagen wir eine Lösung für die Partnersynthese bezüglich schwacher Terminierung und Kostenbeschränktheit vor. Abbildung 48 zeigt abstrakt unser Vorgehen: Wir berechnen im ersten Schritt die kanonische Kostendiskriminante $\delta = \text{CDIS}(L)$ und synthetisieren anschließend einen Partner für L bezüglich $\text{SB}_\mu^\pi[\delta]$.

Sei δ eine Kostendiskriminante. Wir haben bereits in Abschnitt 12.1, S. 195 gezeigt, dass δ die Reduktion der SB_μ^π -Bedienbarkeit auf die $\text{SB}_\mu^\pi[\delta]$ -Bedienbarkeit erlaubt. Wir können die gleichen Argumente anwenden, um Partnersynthese bezüglich SB_μ^π auf Partnersynthese bezüglich $\text{SB}_\mu^\pi[\delta]$ zu reduzieren: Es gilt offenbar

$$\text{SB}_\mu^\pi[\delta][L] \subseteq \text{SB}_\mu^\pi[L] \quad . \quad (100)$$

Ein Algorithmus zur Synthese eines $\text{SB}_\mu^\pi[\delta]$ -Partners löst also auch die Partnersynthese für SB_μ^π .

Im Allgemeinen gilt die Gleichheit der beiden Mengen aus Gleichung (100), S. 197 nicht. Daher lässt sich dieses Ergebnis nicht ohne weiteres auf das Problem der Repräsentation übertragen.

12.3 Kostenoptimale Bedienbarkeit bezüglich schwacher Terminierung

Wie in Abschnitt 7.3.1, S. 155 beobachtet¹, ist die optimale Bedienbarkeit von L bezüglich \succsim_μ^π und \mathbb{S} nicht trivial.

Handelt es sich bei \leq_μ um eine Wohlordnung bis auf ω , können wir mit Hilfe von Satz 215, S. 161 optimale Bedienbarkeit bezüglich \succsim_μ^π auf \mathbb{S} -Bedienbarkeit reduzieren.

Korollar 256

Sei \mathcal{S} eine Klasse von Netzen über \mathbb{K} , für die \mathbb{S} -Bedienbarkeit entscheidbar ist. Sei \leq_μ eine Wohlordnung bis auf ω . Dann ist optimale Bedienbarkeit bezüglich \succsim_μ^π und \mathbb{S} auf \mathcal{S} entscheidbar.

Handelt es sich bei \leq_μ um keine Wohlordnung bis auf ω , können wir Entscheidungsverfahren für \mathbb{S} -Bedienbarkeit und \mathbb{SB}_μ^π -Bedienbarkeit nutzen, um schwächere Aussagen zu treffen:

- (1) Entscheide \mathbb{SB}_μ^π -Bedienbarkeit und \mathbb{S} -Bedienbarkeit von L .
- (2) Sei L nicht \mathbb{SB}_μ^π -bedienbar jedoch \mathbb{S} -bedienbar. Dann ist L laut Korollar 205, S. 156 kostenoptimal \mathbb{S} -bedienbar.
- (3) Sei L weder \mathbb{SB}_μ^π -bedienbar noch \mathbb{S} -bedienbar. Dann ist L trivialerweise nicht kostenoptimal \mathbb{S} -bedienbar.

Ist L jedoch \mathbb{SB}_μ^π -bedienbar, können wir zunächst keine weitere Aussage treffen.

12.4 Synthese und Repräsentation kostenoptimaler Partner bezüglich schwacher Terminierung

Wir betrachten die Probleme der Synthese und Repräsentation eines beziehungsweise aller optimalen Partner bezüglich \succsim_μ^π und \mathbb{S} . Abbildung 49

¹In Abschnitt 7.3.1, S. 155 wird eine beliebige Anforderung ρ betrachtet. In diesem Abschnitt konzentrieren wir uns auf $\rho = \mathbb{S}$.

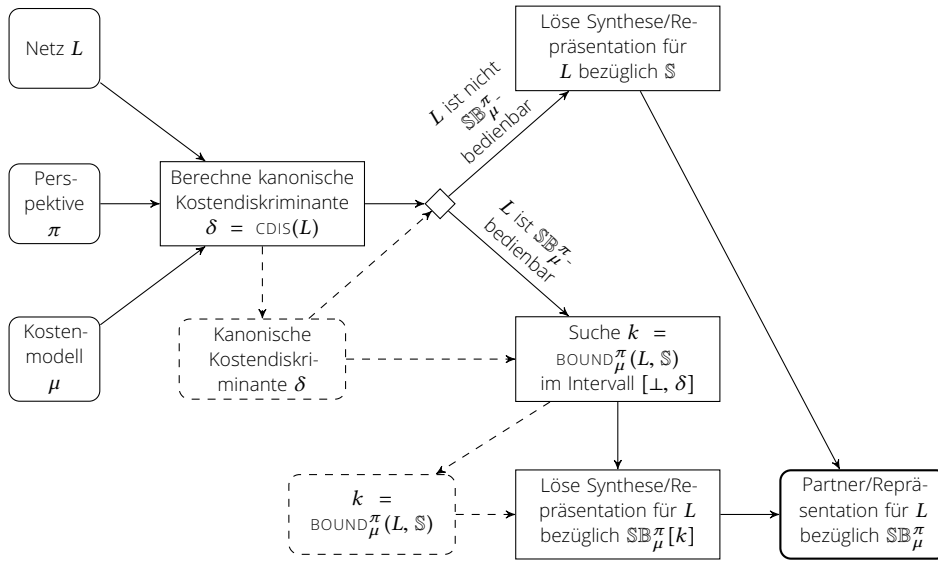


Abbildung 49 Unser Ansatz für Synthese und Repräsentation kostenoptimaler Partner bezüglich schwacher Terminierung

skizziert unseren Ansatz: Wir berechnen die kanonische Kostendiskriminante $\delta = \text{CDIS}(L)$. Ist L SB_{μ}^{π} -bedienbar, suchen wir $k = \text{BOUND}_{\mu}^{\pi}(L, \mathbb{S})$ zwischen dem Bottom-Element \perp und δ und lösen anschließend Partnersynthese/Repräsentation für $\text{SB}_{\mu}^{\pi}[k]$. Ist L nicht SB_{μ}^{π} -bedienbar, lösen wir Partnersynthese/Repräsentation für \mathbb{S} -Bedienbarkeit.

Ist \leq_{μ} eine Wohlordnung bis auf ω , können wir die Synthese (Repräsentation) \succsim_{μ}^{π} -optimaler \mathbb{S} -Partner auf Berechnung einer \mathbb{S} -Kostendiskriminante und Synthese (Repräsentation) bestimmter \mathbb{S} -Partner bzw. $\text{SB}_{\mu}^{\pi}[k]$ -Partner reduzieren. Dies zeigen wir in zwei Schritten. Dabei nutzen wir Lemma 234, S. 178 und Lemma 235, S. 178:

Satz 257 (Reduktion von Partnersynthese und Repräsentation)

Sei \mathcal{S} eine Klasse gewichteter Netze über \mathbb{K} . Sei \leq_{μ} eine Wohlordnung bis auf ω . Sei SB_{μ}^{π} -Bedienbarkeit auf \mathcal{S} entscheidbar. Sei f ein Algorithmus, der zu jedem SB_{μ}^{π} -bedienbaren Netz $L \in \mathcal{S}$ den Wert $\text{BOUND}_{\mu}^{\pi}(L, \mathbb{S})$ berechnet. Seien $\text{SB}_{\mu}^{\pi}[k]$ -Partnersynthese (Repräsentation) und \mathbb{S} -Partnersynthese (Repräsentation) auf \mathcal{S} für alle $k \neq \top \in \mathbb{K}$ lösbar.

Dann ist Partnersynthese (Repräsentation) bezüglich \succsim_{μ}^{π} und \mathbb{S} auf \mathcal{S} lösbar.

Beweis (Satz 257, S. 199)

Sei $L \in \mathcal{S}$. Wir können zunächst \mathbb{SB}_μ^π -Bedienbarkeit von L entscheiden.

- (1) Sei L nicht \mathbb{SB}_μ^π -bedienbar. Laut Lemma 234, S. 178 ist L ein $\{\mathbb{S} \mapsto \mathbb{S}, \succmapsto \text{INDIFF}\}$ -Partner-Repräsentant. Daher fällt Partnersynthese (Repräsentation) für \mathbb{S}, \succ und \mathbb{S} zusammen. Diese sind laut Voraussetzung lösbar auf \mathcal{S} .
- (2) Sei L \mathbb{SB}_μ^π -bedienbar. Laut Lemma 235, S. 178 ist L ein σ' -Partner-Repräsentant für $\sigma = \{\mathbb{S} \mapsto \mathbb{SB}_\mu^\pi[k], \succmapsto \text{INDIFF}\}$ und $k = \text{BOUND}_\mu^\pi(\mathbb{S}, L) = f(L)$. Daher fällt Partnersynthese (Repräsentation) für \mathbb{S}, \succ und $\mathbb{SB}_\mu^\pi[f(L)]$ zusammen. Diese Probleme sind laut Voraussetzung lösbar auf \mathcal{S} .

In Abschnitt 12.1, S. 195 haben wir eine Klasse von Netzen vorgestellt, für die \mathbb{SB}_μ^π -Bedienbarkeit entscheidbar ist. In Abschnitt 10.2.3, S. 186 haben wir Partnersynthese und Repräsentation für $\mathbb{SB}_\mu^\pi[k]$ für ein gegebenes k betrachtet. Daher wenden wir uns im Folgenden der letzten verbleibenden Voraussetzung aus Satz 257, S. 199 zu: Der Berechnung von $\text{BOUND}_\mu^\pi(\mathbb{S}, L)$. Dazu verwenden wir Kostendiskriminanten.

Zunächst zeigen wir den Zusammenhang zwischen einer Kostendiskriminante und $\text{BOUND}_\mu^\pi(L, \mathbb{S})$, der in Abbildung 50 graphisch abgebildet ist:



Abbildung 50 Das Verhältnis zwischen einer Kostendiskriminante δ und $\text{BOUND}_\mu^\pi(L, \rho)$ für kostenbeschränkte L und den Fall, dass \leq_μ eine Wohlordnung bis auf ω ist.

Lemma 258

Sei δ eine Kostendiskriminante von L bezüglich \mathbb{S} . Sei L \mathbb{SB}_μ^π -bedienbar. Sei \leq_μ eine totale Ordnung. Dann gilt:

$$\text{BOUND}_\mu^\pi(L, \mathbb{S}) \leq_\mu \delta .$$

Beweis (Lemma 258)

Da δ eine Kostendiskriminante ist und L \mathbb{SB}_μ^π -bedienbar ist, ist L auch $\mathbb{SB}_\mu^\pi[\delta]$ -bedienbar. Für alle $k < \text{BOUND}_\mu^\pi(L, \mathbb{S})$ gilt, dass L nicht $\mathbb{SB}_\mu^\pi[k]$ -bedienbar ist. Also gilt $\delta \not\leq \text{BOUND}_\mu^\pi(L, \mathbb{S})$. Da \leq_μ eine totale Ordnung ist, gilt die Behauptung.

Da in einer Wohlordnung nur endlich viele k vor einer Kostendiskriminante liegen, können wir aus einer Kostendiskriminante die Schranke $\text{BOUND}_\mu^\pi(\mathbb{S}, L)$ bestimmen, sofern wir $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit für beliebige k entscheiden können.

Lemma 259

Sei \mathcal{S} eine Klasse \mathbb{SB}_μ^π -bedienbarer, gewichteter Netze über \mathbb{K} . Sei f ein Algorithmus, der zu jedem $L \in \mathcal{S}$ eine \mathbb{S} -Kostendiskriminante berechnet. Sei $\mathbb{SB}_\mu^\pi[k]$ -Bedienbarkeit für alle $L \in \mathcal{S}$ und $k \neq \top \in \mathbb{K}$ lösbar. Sei \leq_μ eine Wohlordnung bis auf ω .

Dann ist $\text{BOUNDS}_\mu^\pi(\mathbb{S}, L)$ berechenbar.

Beweis (Lemma 259)

Wir berechnen zunächst $\delta = f(L)$. Da \leq_μ eine Wohlordnung ist und L \mathbb{SB}_μ^π -bedienbar ist, gilt Lemma 258, S. 200. Da \leq_μ eine Wohlordnung ist, liegen vor $f(L)$ nur endlich viele Werte K und $\text{BOUND}_\mu^\pi(\mathbb{S}, L) \in K$. Wir können schrittweise für alle $i \leq_\mu \delta$ $\mathbb{SB}_\mu^\pi[i]$ -Bedienbarkeit entscheiden. Schließlich können wir das kleinste i mit (1) L ist $\mathbb{SB}_\mu^\pi[i]$ -bedienbar und (2) L ist nicht $\mathbb{SB}_\mu^\pi[j]$ bedienbar für alle $j < i$ als $\text{BOUNDS}_\mu^\pi(\mathbb{S}, L)$ zurückgeben.

In Abschnitt 11.2.1, S. 190 haben wir die kanonische Kostendiskriminante definiert (siehe Definition 253, S. 193). Diese wenden wir wie in Abschnitt 12.1, S. 195 an, um den folgenden Satz zu zeigen:

Satz 260 (Lösbarkeit von Partnersynthese und Repräsentation)

Sei $L \oplus \text{MPP}(L)$ beschränkt. Gelte für alle $k \neq \omega \in \mathbb{K}$: (1) $(\mathbb{K}[L, \pi])_{\leq k}^\mu$ ist endlich, und (2) $\omega \notin (\mathbb{K}[L, \pi])_{\leq k}^\mu$. Sei $\mathbb{K}[L, \pi]$ monoton in μ . Sei \leq_μ eine Wohlordnung bis auf ω .

Partnersynthese und Repräsentation für L bezüglich \mathbb{S} und \succ_μ^π ist lösbar.

Beweis (Satz 260)

Laut Korollar 254, S. 196 ist \mathbb{SB}_μ^π -Bedienbarkeit von L entscheidbar. Laut Korollar 247, S. 187 sind Partnersynthese und Repräsentation für L bezüglich $\mathbb{SB}_\mu^\pi[k]$ für beliebige k lösbar. Laut Lemma 259 können wir im Falle der \mathbb{SB}_μ^π -Bedienbarkeit von L auch $\text{BOUND}_\mu^\pi(\mathbb{S}, L)$ berechnen. Die Forderung nach einem Algorithmus f können wir dabei durch die Berechnung von $\text{CDIS}(L)$ erfüllen. Also können wir Satz 257, S. 199 anwenden.

Ist \leq_μ keine Wohlordnung bis auf ω , können wir wie in Abschnitt 12.3, S. 198 folgende, schwächere Aussage treffen: Sei \mathbb{SB}_μ^π -Bedienbarkeit auf \mathcal{S} entscheidbar und $L \in \mathcal{S}$. Ist L nicht \mathbb{SB}_μ^π -bedienbar, können wir \mathbb{S} -Partnersynthese

(\mathbb{S} -Repräsentation) zur Synthese (Repräsentation) \succ_{μ}^{π} -optimaler \mathbb{S} -Partner verwenden.

12.5 Diskussion

Wir gehen zunächst auf die Berechnung der kanonischen Kostendiskriminante ein und vergleichen anschließend den Zusammenhang zwischen den gelösten Problemen mit den Problemen der *Service Selection* und der *Service Composition*.

12.5.1 Berechnung der kanonischen Kostendiskriminante

Die Berechnung der kanonischen Kostendiskriminante $\text{CDIS}(L)$ entspricht dem *Längste-Pfade-Problem*. Dieses ist bekanntermaßen NP-hart, da das *Hamilton-Pfad-Problem* auf das Längste-Pfade-Problem reduziert werden kann [Sch03]. Approximationen und die Abschätzung der Güte der bestmöglichen Approximation sind Gegenstand aktueller Forschung [BHK04; KMR97]. Die Größe des Zustandsraums eines beschränkten Petrinetzes ist zudem exponentiell in der Anzahl seiner Plätze: Sei N ein k -beschränktes Netz mit Platzmenge P , sind höchstens $(k + 1)^{|P|}$ Markierungen erreichbar. Man kann zu jeder Platzmenge P und jedem $k \in \mathbb{N}_0$ außerdem ein Netz N konstruieren mit genau $(k + 1)^{|P|}$ erreichbaren Markierungen und nur $|P|$ Transitionen. Die Größe des most-permissive Partners [Wol09] hängt doppelt-exponentiell von der Größe der Schnittstelle ab. Wenn wir anstatt der kanonischen Kostendiskriminante eine Kostendiskriminante $\delta \geq \text{CDIS}(L)$ berechnen, erhalten wir also eine eher schlechte Approximation von $\text{CDIS}(L)$. Dies kann sich in der weiteren Analyse negativ auswirken, da die Laufzeit der Berechnung von $\text{BOUND}_{\mu}^{\pi}(L, \mathbb{S})$ von der Höhe von δ abhängt. Daher wäre es interessant zu charakterisieren, unter welchen Umständen es sich lohnt, eine präzisere Kostendiskriminante δ zu berechnen, sprich den Abstand zwischen δ und $\text{CDIS}(L)$ zu minimieren, oder wann es besser ist, schnell eine unpräzise Kostendiskriminante zu berechnen.

12.5.2 Zusammenhang mit Service Selection und Service Composition

Das *Service-Selection-Problem* [YL05; Zen+04] beschäftigt sich für einen gegebenen Prozess mit der Frage, welcher Schritt des Prozesses durch welchen Service implementiert werden sollte. Dabei ist ein Prozessmodell und eine Menge von Services gegeben. Das Prozessmodell können wir meist als ein beschriftetes Petrinetz auffassen, wobei eine bestimmte Beschriftung ℓ eine bestimmte Aufgabe ℓ beschreibt. Zu jeder Aufgabe ℓ ist eine Menge von Services gegeben – meist wird der Begriff des Services nicht weiter konkretisiert. Für

jeden Service sind außerdem einige Kennzahlen bekannt, wie beispielsweise die erwartete Laufzeit oder die erwarteten Kosten. Zuletzt ist eine Spezifikation gegeben, zum Beispiel in Form einer zu minimierenden Kostenfunktion. Ziel ist es anschließend, für jede Transition t mit Beschriftung ℓ einen Service aus der Menge der Services von ℓ auszuwählen, sodass der gesamte Prozess bezüglich der Spezifikation optimal ist. Diese Aufgabe ähnelt unserem Problem der Synthese eines optimalen Partners, indem wir die einzelnen Services als Komponenten ansehen. Der Unterschied besteht jedoch darin, dass bei der *Service Selection* die Services als zustandslose Komponenten ohne explizite Interaktion modelliert sind. Es werden also keine Abhängigkeiten des Verhaltens auf Basis von Interaktion modelliert. Wir betrachten Verhalten (und damit auch Kosten), die von der Wahl des Partners abhängig sind. Dies ist bei der *Service Selection* in der Literatur nicht von Belang. Der Fokus liegt in den neueren Forschungsergebnissen vor allem darauf, die Laufzeit der Algorithmen zu reduzieren und vor allem mit Hilfe genetischer Algorithmen so schnell wie möglich eine möglichst gute, nicht zwangsweise optimale Lösung zu finden [TA10; Yan+14]. Wir können das *Service-Selection*-Problem als eine Variante des Problems der Konfiguration [La 08; Aal+10; Got+09] auffassen: Bei der Konfiguration ist für jede Transition im Prozess vor der Ausführung die Wahl zu treffen, ob diese Aktion ausgeführt wird, versteckt wird oder blockiert wird. Wir könnten jede Transition eines Prozessmodells mit Beschriftung ℓ in Transitionen $t_1^\ell \dots t_n^\ell$ zerlegen, wobei jede Transition t_i^ℓ für einen Service von ℓ steht. Dann entspricht eine bestimmte Auswahl von Services einer Konfiguration und wir können den Ansatz aus [ALL12] verwenden, um Konfiguration auf Partnersynthese zu reduzieren.

Beim *Service Composition*-Problem [Cal+08] besteht die Aufgabe darin, einen Service S mit bestimmten Eigenschaften durch *Orchestrierung* aus einer Menge S_1, \dots, S_n gegebener Services zu erzeugen. Orchestrierung bedeutet dabei die Existenz eines *Orchestrators*, der Aufgaben an die anderen Services delegiert. Die Eigenschaft ist dabei häufig selbst als ein Verhaltensmodell gegeben: S soll äquivalent, zum Beispiel bisimilar, zu einem gegebenen Modell M sein. Eine Herangehensweise besteht dann darin, zunächst das Produkt aller S_1, \dots, S_n zu bilden, und anschließend nur solches Verhalten zuzulassen, das in M vorkommt [Ost+12]. Diese Aufgabe kommt dem Orchestrator zu, der jede vom Partner von S kommende Nachricht an einen der S_1, \dots, S_n delegiert und damit auswählt, wer welche Aufgabe erfüllt. Calvanese und Santoso [CS12] betrachten dabei auch Gewichte an Kanten und haben das Ziel, einen optimalen Orchestrator zu berechnen. Obwohl die Aufgabenstellung ähnlich zu unseren Ansätzen ist, treffen wir andere Annahmen: Zum einen ist bei uns die Eigenschaft nicht als ein Modell M sondern als Verknüpfung von Anforderungen und Präferenzen gegeben. Zum anderen hat der Orchestrator volles Wissen über den Zustand der Services $S_1 \dots S_n$. Dies ist bei uns nicht der Fall: Dem Partner ist nur sein Zustand bekannt.

Teil IV

Schluss

In diesem letzten inhaltlichen Teil der Arbeit diskutieren wir unsere Ergebnisse. Wir *evaluieren* unsere Ergebnisse anhand einiger Experimente mit einer prototypischen Implementation in Kapitel 13, S. 207. Anschließend ziehen wir ein *Fazit* in Kapitel 14, S. 217, in welchem wir die Ergebnisse der Arbeit zusammenfassen und klassifizieren, um schließlich mögliche zukünftige Forschungsziele zu identifizieren.

13 Fallstudie

Wir haben einen Ausschnitt der Ergebnisse aus Kapitel 10, S. 181 und Kapitel 12, S. 195 prototypisch im Werkzeug TARA [ST13]¹ implementiert und evaluiert. Wir beschreiben die Implementation in Abschnitt 13.1 und die durchgeführten Experimente in Abschnitt 13.2, S. 209.

13.1 Implementation

Das Werkzeug TARA löst die folgenden Probleme für ein gewichtetes Netz L über \mathbb{N}_0 , die Identitäts-Perspektive, das durch die Max-Plus-Algebra induzierte Kostenmodell und $k \in \mathbb{N}_0$:

- (1) Entscheidung der Bedienbarkeit für \mathbb{SB}_μ^π und $\mathbb{SB}_\mu^\pi[k]$
- (2) Synthese eines Partners für $\mathbb{SB}_\mu^\pi, \mathbb{SB}_\mu^\pi[k]$
- (3) Synthese eines optimalen \mathbb{S} -Partners
- (4) Repräsentation aller Partner für $\mathbb{SB}_\mu^\pi[k]$
- (5) Repräsentation aller optimalen \mathbb{S} -Partner

TARA ist in C++ geschrieben und verfügt über eine Kommandozeilen-Schnittstelle. TARA ist Mitglied einer Werkzeugfamilie [LW10]² zur Analyse und Synthese offener Netze und verwendet in diesem Rahmen folgende Bibliotheken und Werkzeuge:

- Die Bibliothek PNAPI [LW10] zur internen Repräsentation von Netzen;
- das Werkzeug WENDY [LW11a] zur Entscheidung der Bedienbarkeit, der Partnersynthese, der Synthese des most-permissive \mathbb{S} -Partners $\text{MPP}(L)$ und der Repräsentation für \mathbb{S} ;
- das Werkzeug LoLA [Wol07] zur Berechnung des Zustandsraums von $L \oplus \text{MPP}(L)$.

¹Das Werkzeug TARA entstand in Zusammenarbeit mit Marvin Triebel und ist verfügbar unter:
<http://service-technology.org/tara>

²<http://service-technology.org>

Darüber hinaus bindet TARA die Bibliothek LPSOLVE [B+04] zur Lösung linearer Programme ein.

Der Quelltext von TARA setzt die folgenden zwei Kernfunktionalitäten um:

- (1) Konstruktion eines zu L äquivalenten $\langle \mathbb{S}\mathbb{B}_\mu^\pi[k] \mapsto \mathbb{S} \rangle$ -Partner-Repräsentanten für ein gegebenes $k \in \mathbb{N}_0$
- (2) Berechnung der kanonischen Kostendiskriminante $\text{CDIS}(L)$
- (3) Berechnung von $\text{BOUNDS}_\mu^\pi(L, \mathbb{S})$

Die Umsetzung erfolgt jeweils wie in dieser Arbeit beschrieben. Zusätzlich sind für die Berechnung einer Kostendiskriminante mehrere Heuristiken für den Fall eines sehr großen Zustandsraums von $L \oplus \text{MPP}(L)$ verfügbar (siehe auch Abschnitt 12.5, S. 202):

- die Summe über allen Zuständen der jeweils höchsten Kosten einer ausgehenden Transition;
- die Lösung des linearen Programms mit
 - Variablen e_1, \dots, e_n für die n Transitionen des Zustandsraums,
 - Variablen z_1, \dots, z_m für die m Zustände des Zustandsraums,
 - Zielfunktion $\max k_1 e_1 + \dots + k_n e_n$, wobei k_i jeweils den Kosten der Transition e_i entspricht,
 - einer Bedingung $0 \leq e_i \leq 1$ für jede Kante e_i ,
 - einer Bedingung $z = \sum_e$ resultiert in $z^e - \sum_e$ ist aktiviert in z^e für alle Zustände z ,
 - einer Bedingung $-1 \leq z \leq 1$ für den Initialzustand,
 - einer Bedingung $0 \leq z \leq 1$ für alle anderen Zustände,
 - einer Bedingung \sum_z ist ein nicht-initialer Zustand $z = 1$.
- das Produkt aus Anzahl der Zustände und den höchsten an einer beliebigen Transition vorkommenden Kosten.

Dabei wird stets ein Wert berechnet, der über $\text{CDIS}(L)$ liegt – folglich ist der Wert eine Kostendiskriminante. Die minimale Kostenschranke wird nach Ermittlung der Kostendiskriminante in einer Binärsuche ermittelt. Daher ist die Güte der verwendeten Kostendiskriminante nur von sekundärer Bedeutung. Die Binärsuche verwendet als untere Schranke 0 oder eine während der Findung der Kostendiskriminante parallel gefundene untere Schranke k' . Die Schranke k' lässt sich zum Beispiel durch die Berechnung des günstigsten Pfades vom Anfangs- zu einem Zielzustand berechnen: Bei jedem \mathbb{S} -Partner R muss wenigstens ein Ablauf von $L \oplus R$ in einem Zielzustand resultieren.

13.2 Evaluation

In diesem Abschnitt beschreiben wir die Evaluation unseres Prototyps TARA. Ziel der Evaluation ist zu überprüfen, ob die Laufzeit von TARA für einige praktisch relevante Beispiele in einem akzeptablen Bereich liegt.

In Abschnitt 13.2.1 beschreiben wir den Aufbau unserer Experimente. In Abschnitt 13.2.2 betrachten wir die Laufzeit von TARA in Abhängigkeit der Größe der Netze, in Abschnitt 13.2.3, S. 212 in Abhängigkeit der Höhe der Kosten an Transitionen. In Abschnitt 13.2.4, S. 214 vergleichen wir die Laufzeit mit und ohne Berücksichtigung der Kosten. In Abschnitt 13.2.5, S. 215 schließen wir mit einem kurzen Fazit.

13.2.1 Aufbau

Für die Evaluation unseres Ansatzes haben wir die folgenden dreizehn offenen Netze untersucht:

- (1) Akademische Beispiele: *Vending Machine* VM [LMW07], *Olive Oil Ordering* OO [FFK05], *Online Shops* SH3 und SH6 [Loh+06], drei bzw. fünf *Dining Philosophers* DP3 und DP5 adaptiert aus [Dij71],
- (2) Aus der BPEL-Spezifikation [Jor+07] abgeleitete offene Netze: Das *Loan Approval*-Beispiel LN, das *Travel service*-Beispiel in zwei Abstraktionsstufen TR1 und TR2, das *Purchase Order*-Beispiel PO,
- (3) Das Kommunikationsprotokoll SMTP [Pos82]: SMTP.
- (4) Von einer Consulting-Firma (anonymisiert): RE1 und RE2.

Da die aufgezählten Netze nicht gewichtet sind, haben wir diese mit randomisierten Kostenfunktionen über \mathbb{N}_0 untersucht. Dabei haben wir die Kosten aus den drei Intervallen $[0, 1]$, $[0, 10]$ und $[0, 100]$ gewählt, um zu untersuchen, ob die konkrete Höhe der Kosten eine Auswirkung auf die Laufzeit hat. Aus jedem Netz $L \in \{VM, OO, SH3, SH6, DP3, DP5, LN, TR1, TR2, PO, SMTP, RE1, RE2\}$ haben wir neunzig gewichtete Netze wie folgt erzeugt: Für jedes $I \in \mathbb{I}$ mit $\mathbb{I} := \{[0, 1], [0, 10], [0, 100]\}$ erzeugten wir dreißig gewichtete Netze L_I^1, \dots, L_I^{30} mit jeweils zufälligen Kosten aus dem Intervall I .

Daraus resultierten insgesamt $13 \cdot 3 \cdot 30 = 1170$ gewichtete Netze. Für jedes dieser gewichteten Netze L_I^i haben wir mit TARA die folgenden Probleme gelöst:

- (1) DEC: Entscheidung der Bedienbarkeit für \mathbb{SB}_μ^π
- (2) SYN: Synthese eines optimalen \mathbb{S} -Partners
- (3) REP: Repräsentation aller optimalen \mathbb{S} -Partner

Für jedes gewichtete Netz L_I^i und Problem $\text{PROB} \in \{\text{DEC}, \text{SYN}, \text{REP}\}$ ermitteln wir die Laufzeit $\text{PROB}_I^i(L)$ auf demselben Testrechner³.

Die gemessenen Laufzeiten aggregieren wir jeweils mit den Operatoren Minimum, Maximum und arithmetischem Mittel. Als weitere Kennzahlen geben wir die Größe der Schnittstelle (Anzahl $|S|$ der Plätze) sowie des Zustandsraums (Anzahl $|Q|$ der Zustände) des Netzes mit seinem most-permissive \mathbb{S} -Partner an. Die Zeitkomplexität der Berechnung des most-permissive Partners ist doppelt-exponentiell in der Anzahl der Schnittstellenplätze [Wolog]. Die Größe des Zustandsraums beeinflusst vor allem die Berechnung der Kostendiskriminante $\text{cds}(L)$. Wir geben alle Zeiten in Millisekunden an.

13.2.2 Laufzeit in Abhängigkeit der Größe des Netzes

In Tabelle 3 und Tabelle 4 aggregiert eine Zeile die Laufzeiten $\text{PROB}_I^i(L)$ der neunzig aus einem Ausgangsnetz L gebildeten gewichteten Netze L_I^i mit $I \in \mathbb{I}$ und $1 \leq i \leq 30$. Tabelle 3 zeigt die durchschnittliche Laufzeit $\text{PROB}^{\text{avg}}(L)$ – die minimale und maximale Laufzeit $\text{PROB}^{\text{min}}(L)$ und $\text{PROB}^{\text{max}}(L)$ sind in Tabelle 4 aufgeführt.

Netz	S	$ Q $	DEC	SYN	REP
VM	4	14	34	100	101
OO	6	50	43	48	64
SH6	8	77	72	304	305
SH3	7	137	65	283	285
LN	6	156	45	178	179
TR1	8	192	68	191	193
PH3	6	499	68	210	214
TR2	12	1440	929	1144	1149
SMTP	14	1658	1955	2212	2203
RE1	6	2239	439	4224	4225
PO	10	7800	835	1304	1303
RE2	6	27372	3135	8527	8529
PH5	10	43848	22500	23205	23210

Tabelle 3 Durchschnittliche Laufzeit pro Ausgangsnetz

Die Daten aus Tabelle 3 sind in Abbildung 51 visualisiert: Auf der X -Achse tragen wir die Größe $|S|$ der Schnittstelle auf, auf der Y -Achse logarithmisch skaliert die Laufzeit $\text{PROB}^{\text{avg}}(L)$ in Millisekunden.

Für zehn der dreizehn Netze liegt die durchschnittliche Laufzeit $\text{DEC}^{\text{avg}}(L)$ unterhalb einer Sekunde. Für Synthese und Repräsentation liegt die Laufzeit

³DELL PRECISION WORKSTATION T3500 mit Vierkern-CPU INTEL XEON E5507, 4 GiB Hauptspeicher und Betriebssystem UBUNTU 14.04 LTS

Netz	S	Q	DEC		SYN		REP	
			min	max	min	max	min	max
VM	4	14	31	42	53	151	52	147
OO	6	50	31	88	37	107	47	173
SH6	8	77	56	78	177	436	184	448
SH3	7	137	57	70	163	381	178	391
LN	6	156	43	47	118	250	119	239
TR1	8	192	55	72	135	279	132	271
PH3	6	499	58	73	146	262	150	265
TR2	12	1440	909	985	1052	1251	1059	1264
SMTP	14	1658	1906	2079	2091	2372	2085	2405
RE1	6	2239	265	517	1157	6485	1156	6462
PO	10	7800	824	857	1188	1435	1175	1419
RE2	6	27372	3084	3184	6605	10147	6615	10123
PH5	10	43848	22206	22965	22664	23803	22719	23935

Tabelle 4 Minimale und maximale Laufzeit pro Ausgangsnetz

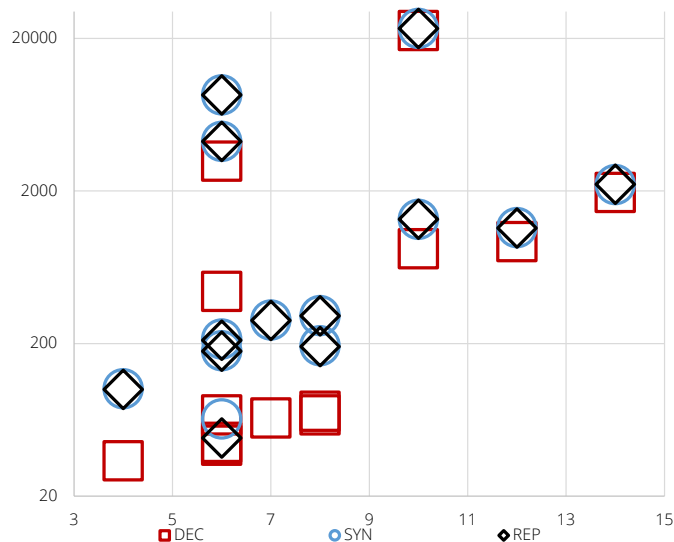


Abbildung 51 Visualisierung der Ergebnisse aus Tabelle 3, S. 210: X-Achse: Anzahl |S| der Schnittstellenplätze, Y-Achse (logarithmisch skaliert): Arithmetisch gemittelte Laufzeit in Millisekunden.

nur für sieben der dreizehn unter einer Sekunde. Die Zehn-Sekunden-Marke knackt für jedes der Probleme einzig PH5, das akademische Beispiel der fünf speisenden Philosophen, das durch den hohen Anteil an Nebenläufigkeit einen sehr großen Zustandsraum erzeugt. Betrachten wir Tabelle 4 fällt uns auf, dass die minimale und maximale Laufzeit jeweils nicht stark voneinander abweichen.

13.2.3 Laufzeit in Abhängigkeit des gewählten Intervalls

In Tabelle 5 betrachten wir die Intervalle $[0, 1]$ und $[0, 100]$ einzeln. Jede Zeile aggregiert die durchschnittlichen Laufzeiten $\text{PROB}_I^i(L)$ der dreißig aus einem Ausgangsnetz L und einem Intervall $I \in \{[0, 1], [0, 100]\}$ gebildeten gewichteten Netze L_I^i mit $1 \leq i \leq 30$. Als Aggregation wählen wir das arithmetische Mittel, Tabelle 5 zeigt also die durchschnittliche Laufzeit $\text{PROB}_I^{\text{avg}}(L)$. In Abbildung 52 visualisieren wir die durchschnittliche Laufzeit für das Problem SYN und alle drei Intervalle $I \in \{[0, 1], [0, 10], [0, 100]\}$.

Netz	$ S $	$ Q $	I	DEC	SYN	REP
VM	4	14	$[0, 1]$	34	67	69
			$[0, 100]$	34	133	134
OO	6	50	$[0, 1]$	41	57	67
			$[0, 100]$	44	43	63
SH6	8	77	$[0, 1]$	69	217	217
			$[0, 100]$	73	391	392
SH3	7	137	$[0, 1]$	63	208	210
			$[0, 100]$	66	353	356
LN	6	156	$[0, 1]$	45	137	138
			$[0, 100]$	46	221	220
TR1	8	192	$[0, 1]$	68	152	153
			$[0, 100]$	68	230	233
PH3	6	499	$[0, 1]$	67	170	176
			$[0, 100]$	68	251	254
TR2	12	1440	$[0, 1]$	924	1091	1104
			$[0, 100]$	927	1194	1190
SMTP	14	1658	$[0, 1]$	1950	2131	2126
			$[0, 100]$	1964	2292	2283
RE1	6	2239	$[0, 1]$	345	2156	2157
			$[0, 100]$	505	6053	6054
PO	10	7800	$[0, 1]$	836	1210	1206
			$[0, 100]$	834	1399	1398
RE2	6	27372	$[0, 1]$	3107	7196	7202
			$[0, 100]$	3153	9785	9787
PH5	10	43848	$[0, 1]$	22460	23131	23115
			$[0, 100]$	22504	23315	23366

Tabelle 5 Durchschnittliche Laufzeit pro Ausgangsnetz und Intervall

Insgesamt können wir feststellen, dass $\text{PROB}_{[0,1]}^{\text{avg}}(L) \leq \text{PROB}_{[0,100]}^{\text{avg}}(L)$ für jedes der Probleme gilt. Eine Ausnahme bildet dabei das Netz OO. Dieses Netz

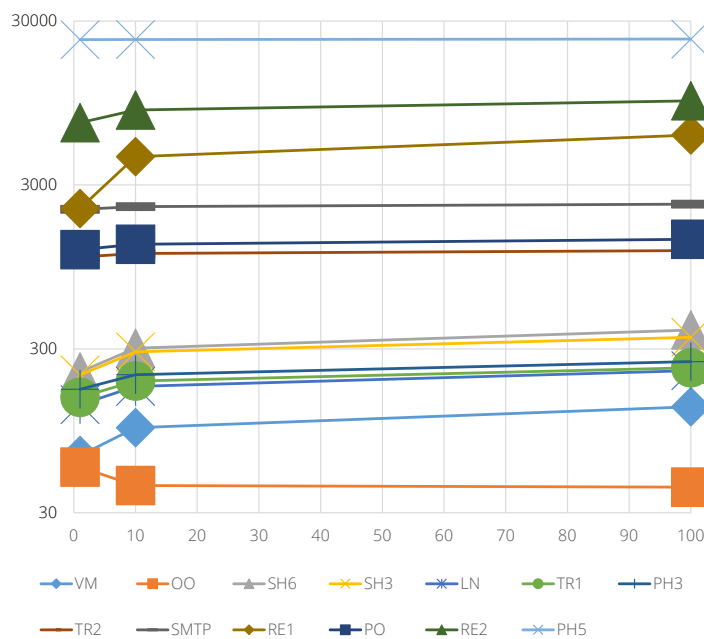


Abbildung 52 Partielle Visualisierung der Ergebnisse aus Tabelle 5, S. 212: X-Achse: Intervall I , Y-Achse (logarithmisch skaliert): Arithmetisch gemittelte Laufzeit des Problems SYN in Millisekunden.

war als einziges für keine der zufälligen Kostenfunktionen kostenbeschränkt bedienbar. Wir vermuten, dass in diesem speziellen Netz höhere Kosten an Transitionen zu einer schnelleren Erkennung der unbeschränkten Kosten führen.

In Tabelle 6 halten wir in jeder Zeile ein Intervall $I \in \mathbb{I}$ fest und aggregieren über allen zu diesem Intervall gebildeten Netzen L_I^i , wobei L ein Ausgangsnetz ist und $1 \leq i \leq 30$, was in den aggregierten Laufzeiten PROB_I^{\min} , $\text{PROB}_I^{\text{avg}}$ und PROB_I^{\max} resultiert. Wir erkennen hier kaum Varianz zwischen den einzelnen Intervallen, was jedoch daran liegen dürfte, dass die Laufzeiten für die einzelnen Ausgangsnetze stark auseinander gehen.

I	DEC			SYN			REP		
	min	avg	max	min	avg	max	min	avg	max
[0, 1]	31	2308	22920	37	2917	23515	47	2918	23734
[0, 10]	32	2329	22965	37	3246	23803	50	3247	23554
[0, 100]	31	2330	22903	38	3512	23752	48	3518	23935

Tabelle 6 Minimale, durchschnittliche und maximale Laufzeit pro Intervall

Daher normieren wir in Tabelle 7 die Laufzeit nach dem Muster:

$$\ell' = c \frac{\ell}{2^{2|S|}} ,$$

wobei ℓ die ursprüngliche und ℓ' die normierte Laufzeit, sowie c eine Konstante ist. Die Wahl dieser Normierung hängt mit der oben beschriebenen Komplexität zur Berechnung des most-permissive Partners zusammen. Wir merken an, dass wir c bei dem Problem DEC anders als bei den Problemen SYN und REP gewählt haben.

I	DEC		SYN		REP	
	avg	max	avg	max	avg	max
$[0, 1]$	3.96	57.13	7.91	123.41	8.04	162.06
$[0, 10]$	4.07	63.71	11.60	169.24	11.71	177.10
$[0, 100]$	3.96	64.14	15.58	229.87	15.75	224.29

Tabelle 7 Durchschnittliche und maximale Laufzeit pro Intervall (normiert)

Abbildung 53 visualisiert die Veränderung der normierten, durchschnittlichen Laufzeiten für die Probleme SYN und REP pro Intervall. Wir können auch hier den ungefähren Trend aus Abbildung 52, S. 213 beobachten.

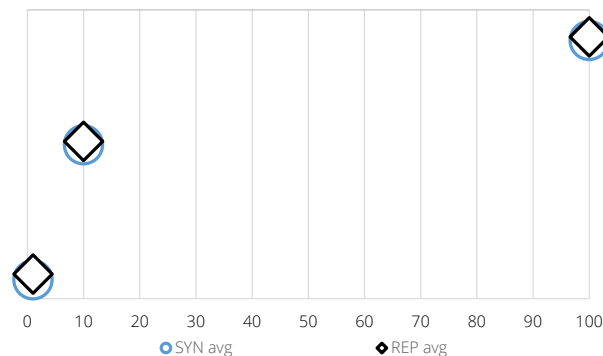


Abbildung 53 Partielle Visualisierung der Ergebnisse aus Tabelle 6, S. 213: X-Achse: Intervall I , Y-Achse (logarithmisch skaliert): Normierte, arithmetisch gemittelte Laufzeit der Probleme SYN und REP in Millisekunden.

13.2.4 Vergleich

Zum Vergleich mit existierenden Methoden für funktionale Verhaltenseigenschaften haben wir zu jedem Netz die Laufzeit des Werkzeugs WENDY zur Entscheidung der \mathbb{S} -Bedienbarkeit, der Synthese des most-permissive \mathbb{S} -Partners

sowie der Berechnung der Repräsentation aller \mathbb{S} -Partner gemessen und den Laufzeiten von TARA gegenübergestellt. Tabelle 8 vergleicht die Laufzeit mit und ohne Betrachtung der Kosten, wobei wir uns auf den Vergleich der durchschnittlichen Laufzeiten beschränken.

Die Lösung der Probleme unter Beachtung von Kosten benötigt deutlich mehr Zeit. Eine einfache Begründung ist, dass für die Lösung aller drei Probleme mit Kosten stets der most-permissive Partner berechnet werden, und anschließend eine Kostendiskriminante aus dem Kompositum mit dem most-permissive Partner berechnet werden muss. Im Gegensatz dazu kann auf die Berechnung des most-permissive Partners bei den Problemen DEC und SYN verzichtet werden, falls man keine Kosten betrachtet.

Netz	S	Q	DEC (avg)		SYN (avg)		REP (avg)	
			mit	ohne	mit	ohne	mit	ohne
VM	4	14	34	11	100	11	101	11
OO	6	50	43	12	48	12	64	12
SH6	8	77	72	22	304	23	305	23
SH3	7	137	65	20	283	20	285	20
LN	6	156	45	13	178	13	179	13
TR1	8	192	68	12	191	14	193	14
PH3	6	499	68	12	210	14	214	14
TR2	12	1440	929	14	1144	30	1149	32
SMTP	14	1658	1955	14	2212	62	2203	63
RE1	6	2239	439	57	4224	57	4225	57
PO	10	7800	835	23	1304	57	1303	57
RE2	6	27372	3135	214	8527	304	8529	304
PH5	10	43848	22500	21	23205	172	23210	176

Tabelle 8 Vergleich der durchschnittlichen Laufzeit mit und ohne Kosten

13.2.5 Fazit

Zusammenfassend können wir sagen, dass bereits eine prototypische Implementierung viele der gestellten Probleme in Laufzeiten unter einer Sekunde lösen kann. Eine Laufzeit von unter einer Sekunde ist unseres Erachtens nach niedrig genug, um TARA beispielsweise aus einem Modellierungs-Werkzeug aufzurufen – entweder auf Knopfdruck oder im Hintergrund bei Änderung des Modells.

Wir können uns jedoch auch viel Potenzial für Optimierung vorstellen: Zur Zeit wird WENDY als eigenständiges Programm (Black Box) verwendet. Der Vorteil ist, dass TARA so wenig wie möglich vorhandene Funktionalität neu implementiert. Der Nachteil ist jedoch, dass einige Daten mehrfach berechnet werden, wie beispielsweise der der most-permissive Partner. Außerdem

ging unseren Beobachtungen zu Folge bei großen most-permissive Partnern sehr viel Zeit bei der Speicherung und dem direkt anschließenden Laden des Partners verloren. Dies könnte sicher eleganter gelöst werden, indem WENDY nicht als reine Black-Box verwendet, sondern möglicherweise eher wie eine Programmbibliothek eingebunden wird, was zum Zeitpunkt der Experimente noch nicht möglich war.

Wir können uns außerdem vorstellen, die Berechnung der Kostendiskriminante bereits während der Berechnung des most-permissive Partners zu beginnen, da zu diesem Zeitpunkt Teile des Zustandsraums von $L \oplus \text{MPP}(L)$ bereits vorliegen. Diese Funktionalität müsste jedoch in den Quelltext von WENDY eingearbeitet werden, oder alternativ sehr viel Funktionalität neu implementiert werden.

14 Fazit

In diesem Kapitel fassen wir die Resultate der Dissertation in Abschnitt 14.1 zusammen, gehen auf die in dieser Arbeit angenommenen Einschränkungen in Abschnitt 14.2, S. 220 ein und zeigen mögliche zukünftige Arbeiten in Abschnitt 14.3, S. 223 auf.

14.1 Zusammenfassung der Ergebnisse

In der Arbeit haben wir uns mit der folgenden zentralen Forschungsfrage beschäftigt: *Wie können wir optimale Partner offener Systeme modellieren, analysieren und synthetisieren?* Der neue Aspekt lag dabei vor allem darin, *nicht-funktionale Aspekte des Verhaltens* zu betrachten. In der Arbeit haben wir diese Aspekte abstrakt als *Kosten* formalisiert

Der resultierende Formalismus der *gewichteten Netze* (Kapitel 3) ermöglicht die formale Modellierung offener Systeme und ihrer Komposition unter Betrachtung funktionaler und nicht-funktionaler Verhaltensaspekte. Auf Basis der Kosten haben wir die Anforderung der *Kostenbeschränktheit* und die Präferenz *kostengünstiger* definiert. Kostenbeschränktheit können wir als eine grundlegende Anforderung an Systeme verstehen, um sicherzustellen, dass Kosten nicht beliebig steigen können. Die Präferenz *kostengünstiger* induziert *kostenoptimale Partner* (Kapitel 7), die die Kosten eines gegebenen offenen Systems minimieren. In Teil II haben wir *Repräsentanten geschlossener Systeme* zur Analyse geschlossener Systeme vorgestellt. Im Mittelpunkt standen dabei die Berechnung der Kosten eines Systems und die Entscheidung der Korrektheit bezüglich Kostenbeschränktheit (Kapitel 5). Die dabei entwickelten Konzepte haben wir in Teil III angewendet, um *Repräsentanten und Kostendiskriminanten offener Systeme* (Kapitel 9 und 11) zu definieren. Diese Konzepte haben wir in Kapitel 10 und 12 zur Analyse und Synthese offener Systeme verwendet. Dabei standen die Entscheidung der kostenbeschränkten Bedienbarkeit sowie die Synthese optimaler Partner im Vordergrund.

Unsere Beiträge haben wir im *einheitlichen Formalismus* der gewichteten Netze formuliert. Die Techniken zur Analyse und Synthese sind mehrheitlich *Reduktionen auf bestehende Techniken*. Auf diese Weise ist es möglich, bereits bestehende Forschungsergebnisse und Werkzeuge aus der Analyse- und Synthese wiederzuverwenden.

14.1.1 Formale Modellierung offener Systeme mit gewichteten Netzen

Gewichtete Netze decken sowohl funktionale wie nicht-funktionale Verhaltensaspekte ab:

- *Gewichtete Netze sind besondere Petrinetze.* Der Formalismus der Petrinetze ist etabliert in der formalen Modellierung, Analyse, Verifikation und Synthese verteilter Systeme. Alle existierenden Techniken sind sofort auf gewichtete Netze anwendbar.
- *Gewichtete Netze modellieren nicht-funktionale Aspekte durch Kosten.* Mit dem abstrakten Begriff der Kosten können wir nicht-funktionale Aspekte des Verhaltens wie den Energie-, Zeit- oder sonstige Ressourcenverbrauch oder die Zuverlässigkeit eines Systems modellieren. Diese abstrakte Herangehensweise ist erprobt in der Theorie der *gewichteten Automaten*.

Wir stützen demnach unseren Formalismus zur Modellierung offener Systeme auf existente Formalismen. Wir unterscheiden im Gegensatz zu der Theorie der gewichteten Automaten zwischen Perspektiven: Eine Perspektive erlaubt die Auswahl oder Transformation von Abläufen. Diese Betrachtung von Kosten ist genereller als die in gewichteten Automaten: In gewichteten Automaten werden die Kosten stets für ein Wort definiert, das durch die Kantenbeschriftungen und Endzustände vom Automaten erzeugt oder akzeptiert wird. Diese Generalisierung erlaubt unter anderem eine Trennung von Belangen und die Definition lokaler, also zu einer Komponente gehörender Kosten. Wir untersuchen den Zusammenhang zwischen einigen exemplarischen Perspektiven und zeigen, dass die Filter-Perspektiven als eine grundlegende Perspektive hervorstechen: Techniken zur Analyse bezüglich der Filter-Perspektiven lassen sich mit Hilfe kanonischer Konstruktionen auch für andere Perspektiven verwenden. Die von uns betrachtete Klasse von Kostenmodellen weicht insofern von den in der Theorie der gewichteten Automaten verwendeten Klassen ab, dass wir einen großen Fokus darauf legen, mit unendlich vielen Kostenwerten umzugehen: Ein verteiltes System hat im Allgemeinen unendlich viele Abläufe, die jeweils unterschiedliche Kosten erzeugen können.

14.1.2 Formale Modellierung der Korrektheit und Optimalität bezüglich nicht-funktionaler Verhaltensaspekte

In dieser Arbeit stellen wir die nicht-funktionale Verhaltens-Anforderung der Kostenbeschränktheit sowie die Präferenz *kostengünstiger* vor:

- Die Anforderungen der k -Kostenbeschränktheit für gegebene k und die Kostenbeschränktheit fordern eine grundlegende *Korrektheit bezüglich Kosten*: Die Kosten eines Systems liegen unter einer Schranke.

- Die Präferenz *kostengünstiger* erlaubt kanonisch eine *Ordnung von Systemen auf Basis von Kosten*. Als Parameter dient uns dabei eine gegebene Präferenzrelation. Wir haben uns dabei für eine Modellierung als vollständiger Verband entschieden und Kostenmodelle identifiziert, die solch eine Ordnung induzieren.

Dabei parametrisieren wir unsere Anforderungen und Präferenzen stets mit Perspektiven und Kostenmodellen. So ist es möglich, beispielsweise die Kostenbeschränktheit bezüglich einer bestimmten Kombination aus Perspektive und Kostenmodell zu fordern und bezüglich einer anderen Kombination von Perspektive und Kostenmodell die Systeme zu ordnen.

14.1.3 Analyse und Synthese offener Systeme bezüglich nicht-funktionaler Verhaltensaspekte

Wir lösen die folgenden Probleme für Unterklassen von Perspektiven und Kostenmodellen:

- Entscheidung der Korrektheit bezüglich der Anforderungen der k -Kostenbeschränktheit und der Kostenbeschränktheit durch Reduktion auf Beschränktheit und Erreichbarkeit.
- Entscheidung der Bedienbarkeit sowie Lösung der Partnersynthese bezüglich der Anforderung der k -Kostenbeschränktheit bzw. Kostenbeschränktheit durch Reduktion auf Korrektheit bezüglich k -Kostenbeschränktheit bzw. Kostenbeschränktheit.
- Entscheidung der Bedienbarkeit sowie Lösung der Partnersynthese und Repräsentation bezüglich der Konjunktion der Anforderungen schwache Terminierung und k -Kostenbeschränktheit durch Reduktion auf das entsprechende Problem bezüglich schwacher Terminierung.
- Entscheidung der Bedienbarkeit sowie Lösung der Partnersynthese bezüglich der Konjunktion der Anforderungen der schwachen Terminierung und der Kostenbeschränktheit durch die Berechnung eines speziellen Kostenwertes k und anschließende Reduktion auf das entsprechende Problem bezüglich schwacher Terminierung und k -Kostenbeschränktheit.
- Entscheidung der kostenoptimalen Bedienbarkeit bezüglich der Anforderung der schwachen Terminierung durch Reduktion auf Entscheidung der Bedienbarkeit bezüglich schwacher Terminierung.
- Synthese und Repräsentation kostenoptimaler Partner bezüglich der Anforderung der schwachen Terminierung durch Berechnung eines

kanonischen, minimalen k und anschließende Reduktion auf das entsprechende Problem bezüglich schwacher Terminierung und k -Kostenbeschränktheit.

Die grundlegende Lösungstechnik in dieser Arbeit sind die in Abschnitt 1.5, S. 12 beschriebenen Repräsentanten. Im einzelnen konstruieren wir:

- Perspektiven-Repräsentanten zur Reduktion des Parameters der Perspektive auf Filter-Perspektiven,
- Wert-Repräsentanten zur erleichterten Berechnung der Kosten eines Systems und der Entscheidung der k -Kostenbeschränktheit eines Systems,
- Schranken-Repräsentanten zur erleichterten Entscheidung der Kostenbeschränktheit eines Systems,
- kompositionale Perspektiven-Repräsentanten, Wert-Repräsentanten sowie Schranken-Repräsentanten zur Betrachtung von Komposita,
- Partner-Repräsentanten zur erleichterten Entscheidung der Bedienbarkeit sowie Lösbarkeit der Partnersynthese und Repräsentation.

Zusammenfassend haben wir Analyse- und Syntheseprobleme bezüglich funktionaler und nicht-funktionaler Verhaltensaspekte durch Reduktion auf bekannte Techniken gelöst. Im Werkzeug TARA haben wir einen Großteil dieser Probleme am Beispiel des Kostenmodells der Max-Plus-Algebra prototypisch implementiert und in einer kurzen Fallstudie diesen Prototypen evaluiert.

14.2 Einschränkungen und offene Probleme

In dieser Arbeit haben wir drei einschränkende Entscheidungen getroffen:

- (1) Wir haben schwache Terminierung als grundlegende funktionale Verhaltens-Anforderung gewählt.
- (2) Wir haben bevorzugt Verbands-Kostenmodelle betrachtet.
- (3) Wir haben in der gesamten Arbeit vom Einfluss nicht-funktionaler Aspekte auf funktionale Aspekte des Verhaltens abstrahiert.

Wir gehen in den folgenden Unterabschnitten auf diese drei Einschränkungen ein und zeigen die damit verbundenen offenen Probleme auf.

14.2.1 Schwache Terminierung als grundlegende funktionale Verhaltens-Anforderung

In dieser Arbeit haben wir schwache Terminierung als grundlegende funktionale Verhaltens-Anforderung betrachtet. Schwache Terminierung ist zum einen bekannt als Soundness im Bereich der formalen Modellierung und Analyse von Geschäftsprozessen. Zum anderen ist sie eine Kombination aus Sicherheits- und Lebendigkeitseigenschaft: Die Forderung, dass jede erreichbare Markierung eine Rahmenmarkierung ist, ist eine Sicherheitseigenschaft; die Forderung nach der Erreichbarkeit einer Zielmarkierung ist eine Lebendigkeitseigenschaft. Diese Formulierung von Korrektheit durch zwei charakteristische Markierungsmengen ist natürlich bei weitem nicht so aussagemächtig wie temporale Logiken, allerdings lassen sich mit Rahmenmarkierungen einfache Sicherheitseigenschaften im Stil *es ist kein schlechter Zustand erreichbar* formulieren, und mit Hilfe von Zielmarkierungen können erwünschte Zustände ausgedrückt werden. Soll nur einer der Aspekte (Sicherheit bzw. Lebendigkeit) betrachtet werden, kann eine der beiden Mengen universell bzw. leer gewählt werden. Schließlich ist schwache Terminierung eine Anforderung, für die bereits Analyse- und Synthesemethoden existieren. Dies ist vor allem aus dem Grund von Nutzen, dass wir in dieser Arbeit vorwiegend Reduktionen nicht-funktionaler Aspekte auf funktionale Aspekte vornehmen.

Es ist eine offene Frage, welche der Ergebnisse dieser Arbeit direkt auf andere funktionale Verhaltens-Anforderungen übertragen werden können. Wir können die folgenden drei notwendigen Bedingungen für die Anwendbarkeit unserer Ergebnisse angeben:

- Die Existenz eines endlichen *most-permissive Partners* ist ein grundlegender Bestandteil unserer Überlegungen für die Berechnung von Kostendiskriminanten.
- Die Austauschbarkeit bisimilarer Systeme, das heißt, sind $L \oplus \text{Env}(L)$ und $L' \oplus \text{Env}(L')$ bisimilar, haben L und L' auch dieselben Partner. Dies ist für alle Eigenschaften der *Computation Tree Logic* erfüllt.
- Die Möglichkeit, unerwünschte Markierungen zu formulieren, nutzen wir bei der Konstruktion von Partner-Repräsentanten aus Schranken-Repräsentanten aus, indem wir solche Markierungen als unerwünscht kennzeichnen, in denen die Kosten eine gegeben. Bei einer Erweiterung auf die *Computation Tree Logic* wäre das trivialerweise gegeben, da wir diese Markierungen per Konjunktion anhängen können.

Insgesamt sind wir optimistisch, dass die Techniken dieser Arbeit auch auf andere funktionale Verhaltens-Anforderungen zu übertragbar sind.

14.2.2 Verbands-Kostenmodelle als grundlegende Kostenmodelle

Wir haben in dieser Arbeit vorwiegend Verbands-Kostenmodelle anstatt beliebiger Kostenmodelle betrachtet. Motivation dafür war, dass Verbands-Kostenmodelle inhärent kompositional sind und daher den Rückschluss von den erreichbaren Markierungen auf die Kosten eines Systems erlauben. Es ist anzunehmen, dass Verbands-Kostenmodelle nicht die größte interessante Klasse kompositionaler Kostenmodelle sind: So ist beispielsweise jedes durch einen partiell vollständigen Multi-Hemiring induzierte Kostenmodell kompositional. Es wäre ebenso möglich, durch die geschickte Wahl einer Perspektive, die Menge der zu betrachtenden Kostenwerte auf eine endliche Menge zu beschränken.

Verbands-Kostenmodelle erleichtern die Betrachtung von Problemen, die den Vergleich unterschiedlicher Komposita von Systemen behandeln: In einem Verbands-Kostenmodell können durch Komposition höchstens mehr, jedoch niemals weniger Kosten entstehen. Wie in Abschnitt 3.8.5, S. 96 beschrieben, eignen sich vollständige Verbände gut zur Modellierung des *Worst Case* bei unendlich vielen möglichen Ausgängen eines Szenarios. Insofern halten wir Verbands-Kostenmodelle für einen ersten Schritt in die richtige Richtung. Ein interessantes offenes Problem wäre die Frage, welche in dieser Arbeit entwickelten Techniken auch für andere Kostenmodelle adaptiert werden können.

14.2.3 Abstraktion vom Einfluss nicht-funktionaler Aspekte auf funktionale Aspekte

In dieser Arbeit haben wir funktionale und nicht-funktionale Anforderungen und Präferenzen gemeinsam betrachtet. Dabei hatten funktionale Aspekte des Verhaltens stets direkten Einfluss auf die nicht-funktionalen Aspekte: Wir haben die Wert-Semantik eines Systems auf Basis der Menge seiner Abläufe gebildet. Wir sind jedoch stets davon ausgegangen, dass nicht-funktionale Aspekte keinen Einfluss auf die funktionalen haben. Hierbei handelt es sich insofern um eine Einschränkung, dass wir uns gut vorstellen können, dass die Entwicklung der Kosten in der Realität beispielsweise durchaus einen Einfluss auf das weitere Verhalten haben können. Wir gehen in der Arbeit davon aus, dass solche Einschränkungen der möglichen Abläufe bereits im Modell enthalten sind. Dies wäre beispielsweise dadurch möglich, dass die Kosten, die das Verhalten beeinflussen, wie in Abschnitt 3.5, S. 70 beschrieben in das Verhalten kodiert werden. Damit grenzen wir uns gleichzeitig scharf von Formalismen ab, die eben solche Einschränkungen explizit formulieren, wie beispielsweise Timed Automata oder Zeitpetrinetze (siehe auch Abschnitt 3.8.1, S. 92). So erreichen wir es, dass wir unsere Ergebnisse auf ein ausdrucks mächtiges Modell

mit zahlreichen positiven Entscheidbarkeitsresultaten wie Erreichbarkeit oder Beschränktheit stützen können.

14.3 Zukünftige Arbeiten

Im vorherigen Abschnitt 14.2, S. 220 haben wir bereits einige offene Fragen aufgeworfen. In diesem Abschnitt diskutieren wir einige weitere Möglichkeiten für zukünftige Arbeiten.

14.3.1 Globale oder dezentrale Betrachtung der Kosten

In dieser Arbeit haben wir Kosten stets lokal für genau eine Komponente betrachtet. Es war dann Aufgabe eines Partners, die Kosten zu minimieren. Darüber hinaus wären folgende Betrachtungsweisen möglich:

- *Globale Sicht*: Wir betrachten die Gesamtkosten eines Kompositums durch *Aggregation der Kosten aller Komponenten*.
- *Dezentrale Sicht*: Wir betrachten *lokal die Kosten mehrerer Komponenten*.

Bei der globalen Sicht besteht die Herausforderung darin, Kosten über die Grenzen von Komponenten hinweg zu aggregieren. Bei der dezentralen Sicht besteht die Herausforderung darin, die möglicherweise gegensätzlichen Interessen mehrerer Komponenten gleichzeitig zu betrachten. Hier wäre die Anwendung von Konzepten aus der Spieltheorie interessant – ein optimaler Partner könnte beispielsweise einzelne Komponenten bevorzugen oder versuchen, ein *Nash-Gleichgewicht* [Nas51] sicherstellen.

14.3.2 Erstellung von Modellen

In dieser Arbeit haben wir Modelle in Form gewichteter Netze als gegeben angenommen. Der Formalismus der Petrinetze ist, wie in Abschnitt 3.8.1, S. 92 beschrieben, im Bereich der Modellierung offener Systeme etabliert und es existieren zahlreiche Verbindungen zu Sprachen aus der Industrie. Daher sind wir optimistisch, dass es möglich ist, auch in der Realität Modelle in Form gewichteter Netze zu erstellen. Eine erste Idee wäre, die Kosten einer Transition aus Aufzeichnungen über die Ausführung bestehender Systeme abzuleiten. Als ein Beispiel enthalten *Event Logs* aus der in der Praxis angewendeten Technik des *Process Mining* [Aal+12] bereits Informationen über die Ausführung einer Transition, wie beispielsweise die verstrichene Zeitdauer [AD02]. Darüber hinaus sind in der Industrie angewendete Sprachen häufig erweiterbar. Wir können uns beispielsweise gut vorstellen, die Sprache BPMN [CT12] sowie ihre

Petrinetz-Semantik [DDO08b] so anzupassen, dass Kosten modelliert werden können.

14.3.3 Betrachtung von Wahrscheinlichkeiten

In dieser Arbeit haben wir gewichtete Netze als besondere Petrinetze modelliert. Dementsprechend haben wir keinerlei Aussagen über die Wahrscheinlichkeit für das Eintreten eines bestimmten Ablaufs getroffen. Wir können daher zunächst erst einmal keine Aussage über den *Average Case*, also das durchschnittliche oder typische Verhalten eines Systems treffen. Stochastische Petrinetze [Kar+94] erlauben die Modellierung von Wahrscheinlichkeiten für das Eintreten bestimmter Transitionen und damit auch für gesamte Abläufe. Damit wäre es möglich, den *Average Case* zu betrachten, indem die Kosten entsprechend der Eintrittswahrscheinlichkeiten der Transitionen aggregiert werden. Dadurch wird das Modell jedoch deutlich komplexer und die bisherigen Techniken sind nicht ohne weiteres anwendbar.

14.3.4 Betrachtung von Daten

Bisher betrachten wir nur den Kontrollfluss eines Systems in Form von Zustandsübergängen und abstrahieren von Nachrichteninhalten und Daten-abhängigem Verhalten. Wir können uns jedoch gut vorstellen, dass die Kosten einer Transition von den verarbeiteten Daten abhängen und von Eintreten zu Eintreten der Transition auf Basis der Daten variieren können. Solange es sich bei diesen Abhängigkeiten nur um endlich viele verschiedene Fälle handelt, können wir möglicherweise endlich viele Äquivalenzklassen von Datenwerten bilden oder statische Analyse-Techniken anwenden [HAM14]. Ist dies nicht möglich, können wir uns *symbolische Techniken* zur Lösung dieser Problematik vorstellen [LW11b; Wag12].

Appendix

Literatur

- [Aal+08] W. M. P. van der Aalst, N. Lohmann, P. Massuthe, C. Stahl und K. Wolf. „From Public Views to Private Views – Correctness-By-Design for Services“. In: *Lecture Notes in Computer Science 4937*. Hrsg. von M. Dumas und R. Heckel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, S. 139–153 (siehe S. 9, 179).
- [Aal+09] W. M. P. van der Aalst, N. Lohmann, P. Massuthe, C. Stahl und K. Wolf. „Multiparty Contracts: Agreeing and Implementing Interorganizational Processes“. In: *The Computer Journal* 53.1 (2009), S. 90–106 (siehe S. 9).
- [Aal+10] W. M. P. van der Aalst u. a. „Preserving Correctness During Business Process Model Configuration“. In: *Formal Aspects of Computing* (2010) (siehe S. 9, 203).
- [Aal+11] W. M. P. van der Aalst u. a. „Soundness of Workflow Nets: Classification, Decidability, and Analysis“. In: *Formal Aspects of Computing* 23.3 (2011), S. 333–363 (siehe S. 8).
- [Aal+12] W. M. P. van der Aalst u. a. „Process Mining Manifesto“. In: *Lecture Notes in Business Information Processing 99*. Hrsg. von F. Daniel, K. Barkaoui und S. Dustdar. Springer Berlin Heidelberg, 2012, S. 169–194 (siehe S. 93, 223).
- [Aal02] W. M. van der Aalst. „Making Work Flow: On the Application of Petri Nets to Business Process Management“. In: *Application and Theory of Petri Nets 2002*. Hrsg. von J. Esparza und C. Lakos. Bd. 2360. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, S. 1–22 (siehe S. 93).
- [Aal97] W. M. P. van der Aalst. „Verification of Workflow Nets“. In: *Application and Theory of Petri Nets 1997*. Hrsg. von P. Azéma und G. Balbo. Bd. 1248. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1997, S. 407–426 (siehe S. 8).
- [AD02] W. M. P. van der Aalst und B. F. van Dongen. „Discovering Workflow Performance Models from Timed Logs“. In: *Lecture Notes in Computer Science 2480*. Hrsg. von Y. Han, S. Tai und D. Wikarski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, (siehe S. 223).
- [AD94] R. Alur und D. L. Dill. „A Theory of Timed Automata.“ In: 1994, S. 183–235 (siehe S. 93).
- [ALL12] W. M. P. van der Aalst, N. Lohmann und M. La Rosa. „Ensuring Correctness During Process Configuration Via Partner Synthesis“. In: *Information Systems* 37.6 (2012), S. 574–592 (siehe S. 9, 203).
- [Arm+10] M. Armbrust u. a. „A View of Cloud Computing“. In: *Commun. ACM* 53.4 (2010), S. 50–58 (siehe S. 3).
- [AV03] A. Arnold und A. Vincent. „Games for Synthesis of Controllers with Partial Observation“. In: *Theoretical Computer Science* 303.1 (2003), S. 7–34 (siehe S. 8, 188).
- [B+04] M. Berkelaar, K. Eikland, P. Notebaert u. a. „Lpsolve: Open Source (mixed-Integer) Linear Programming System“. In: *Eindhoven U. of Technology* (2004) (siehe S. 208).
- [Bae05] J. C. Baeten. „A Brief History of Process Algebra“. In: *Theoretical Computer Science* 335.2 (2005), S. 131–146 (siehe S. 3, 93).

- [BBCo5] A. Bracciali, A. Brogi und C. Canal. „A Formal Approach to Component Adaptation“. In: *J. Syst. Softw.* 74.1 (2005), S. 45–54 (siehe S. 9).
- [Ben+05] B. Benatallah, F. Casati, D. Grigori, H. Nezhad und F. Toumani. „Developing Adapters for Web Services Integration“. In: *Advanced Information Systems Engineering*. Hrsg. von O. Pastor und J. Falcão e Cunha. Bd. 3520. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, S. 415–429 (siehe S. 9).
- [Ben+10] S. Bensalem, M. Bozga, S. Graf, D. Peled und S. Quinton. „Methods for Knowledge Based Controlling of Distributed Systems“. In: *Automated Technology for Verification and Analysis* (2010), S. 52–66 (siehe S. 8, 188).
- [BHKo4] A. Björklund, T. Husfeldt und S. Khanna. „Approximating Longest Directed Paths and Cycles“. In: *Automata, languages and programming*. Bd. 3142. Lecture Notes in Comput. Sci. Springer, Berlin, 2004, S. 222–233 (siehe S. 202).
- [BKo8] C. Baier und J.-P. Katoen. *Principles of Model Checking*. Bd. 26202649. MIT press Cambridge, 2008 (siehe S. 33).
- [BKP92] F. S. de Boer, J. W. Klop und C. Palamidessi. „Asynchronous Communication in Process Algebra“. In: *Logic in Computer Science, 1992. LICS '92., Proceedings of the Seventh Annual IEEE Symposium on*. 1992, S. 137–147 (siehe S. 93).
- [Blo+09] R. Bloem, K. Chatterjee, T. A. Henzinger und B. Jobstmann. „Better Quality in Synthesis Through Quantitative Objectives“. In: *Computer Aided ...* (2009) (siehe S. 8, 188).
- [BPM83] M. Ben-Ari, A. Pnueli und Z. Manna. „The Temporal Logic of Branching Time“. In: *Acta Informatica* 20.3 (1983), S. 207–226 (siehe S. 95).
- [BR88] J. J. Berstel und C. Reutenauer. *Rational Series and Their Languages*. New York, NY, USA: Springer-Verlag New York, Inc., 1988 (siehe S. 129).
- [Bri+11] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini und J. F. Raskin. „Faster Algorithms for Mean-Payoff Games“. In: *Form. Methods Syst. Des.* 38.2 (2011), S. 97–118 (siehe S. 126).
- [BW90] J. Baeten und W. P. Weijland. *Process Algebra, Volume 18 of Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press Cambridge, UK, 1990 (siehe S. 93).
- [Cal+08] D. Calvanese, G. De Giacomo, M. Lenzerini, M. Mecella und F. Patrizi. „Automatic Service Composition and Synthesis: The Roman Model.“ In: *IEEE Data Eng. Bull.* 31.3 (2008), S. 18–22 (siehe S. 94, 203).
- [CDHo8] K. Chatterjee, L. Doyen und T. A. Henzinger. „Quantitative Languages“. In: *Computer Science Logic*. Hrsg. von M. Kaminski und S. Martini. Bd. 5213. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, S. 385–400 (siehe S. 95, 129).
- [CGP01] E. M. Clarke, O. Grumberg und D. Peled. *Model Checking*. MIT Press, 2001 (siehe S. 3).
- [Cie+88] R. Cieslak, C. Desclaux, A. Fawaz und P. Varaiya. „Supervisory Control of Discrete-Event Processes with Partial Observations“. In: *Automatic Control, IEEE Transactions on* 33.3 (1988), S. 249–260 (siehe S. 8).
- [CKo7] T. Chothia und J. Kleijn. „Q-Automata: Modelling the Resource Usage of Concurrent Components“. In: *Electronic Notes in Theoretical Computer Science* 175.2 (2007), S. 153–167 (siehe S. 94).
- [Cou96] P. Cousot. „Abstract Interpretation“. In: *Symposium on Models of Programming Languages and Computation, ACM Computing Surveys* 28.2 (1996), S. 324–328 (siehe S. 126).

- [CS12] D. Calvanese und A. Santoso. „Best Service Synthesis in the Weighted Roman Model“. In: *Proc. of the 4th Central-European Workshop on Services and their Composition (ZEUS 2012)*. Bd. 847. CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/>. 2012, S. 42–49 (siehe S. 94, 203).
- [CT12] M. Chinosi und A. Trombetta. „BPMN: An Introduction to the Standard“. In: *Computer Standards & Interfaces* 34.1 (2012), S. 124–134 (siehe S. 92, 223).
- [DDOo8a] R. M. Dijkman, M. Dumas und C. Ouyang. „Semantics and Analysis of Business Process Models in BPMN“. In: *Information and Software Technology* 50.12 (2008), S. 1281–1294 (siehe S. 92).
- [DDOo8b] R. M. Dijkman, M. Dumas und C. Ouyang. „Semantics and Analysis of Business Process Models in Bpmn“. In: *Information and Software technology* 50.12 (2008), S. 1281–1294 (siehe S. 224).
- [Dij71] E. W. Dijkstra. „Hierarchical Ordering of Sequential Processes“. In: *Acta Inf.* 1 (1971), S. 115–138 (siehe S. 209).
- [DK09] M. Droste und W. Kuich. „Semirings and Formal Power Series“. In: *Handbook of Weighted Automata*. Hrsg. von M. Droste, W. Kuich und H. Vogler. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2009, S. 3–28 (siehe S. 129).
- [DK13] M. Droste und W. Kuich. „Weighted Finite Automata Over Hemirings“. In: *Theoretical Computer Science* 485 (2013), S. 38–48 (siehe S. 21, 96, 129, 130).
- [DKV09] M. Droste, W. Kuich und H. Vogler. *Handbook of Weighted Automata*. 1st. Springer Publishing Company, Incorporated, 2009 (siehe S. 93, 129).
- [DM11] M. Droste und I. Meinecke. „Weighted Automata and Regular Expressions Over Valuation Monoids“. In: *International Journal of Foundations of Computer Science* 22.08 (2011), S. 1829–1844 (siehe S. 21, 46, 95).
- [DM12] K. Duske und R. Müller. „A Survey on Approaches for Timed Services.“ In: *ZEUS*. 2012, S. 66–73 (siehe S. 94).
- [Don+04] X. Dong, A. Halevy, J. Madhavan, E. Nemes und J. Zhang. „Similarity Search for Web Services“. In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment. 2004, S. 372–383 (siehe S. 9).
- [Dor32] J. L. Dorroh. „Concerning Adjunctions to Algebras“. In: *Bulletin of the American Mathematical Society* 38.2 (1932), S. 85–88 (siehe S. 130).
- [DSWo6] M. Dumas, M. Spork und K. Wang. „Adapt or Perish: Algebra and Visual Notation for Service Interface Adaptation“. In: *Business Process Management*. Hrsg. von S. Dustdar, J. Fiadeiro und A. Sheth. Bd. 4102. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, S. 65–80 (siehe S. 9).
- [Dzi+14] R. Dzikowski, A. Moelle, W. Reisig und J. Sürmeli. *Asynchronous Communication*. Techn. Ber. 246. 2014 (siehe S. 139, 162).
- [Eil74] S. Eilenberg. *Automata, Languages, and Machines*. Orlando, FL, USA: Academic Press, Inc., 1974 (siehe S. 129).
- [ÉK09] Z. Ésik und W. Kuich. „Finite Automata“. In: *Handbook of Weighted Automata*. Hrsg. von M. Droste, W. Kuich und H. Vogler. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2009, S. 69–104 (siehe S. 95).
- [FFK05] J. A. Fisteus, L. S. Fernández und C. D. Kloos. „Applying Model Checking to Bpel4ws Business Collaborations“. In: *Proceedings of the 2005 ACM symposium on Applied computing*. SAC ’05. Santa Fe, New Mexico: ACM, 2005, S. 826–830 (siehe S. 209).

- [Fin93] A. Finkel. „The Minimal Coverability Graph for Petri Nets“. In: *Advances in Petri Nets 1993*. Hrsg. von G. Rozenberg. Bd. 674. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1993, S. 210–243 (siehe S. 117).
- [Fro04] J. Fromm. *The Emergence of Complexity*. Kassel university press Kassel, 2004 (siehe S. 3).
- [FV09] Z. Fülöp und H. Vogler. „Weighted Tree Automata and Tree Transducers“. In: *Handbook of Weighted Automata*. Hrsg. von M. Droste, W. Kuich und H. Vogler. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2009, S. 313–403 (siehe S. 95).
- [Gam+94] E. Gamma, R. Helm, R. Johnson und J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1994 (siehe S. 3).
- [GMW12] C. Gierds, A. J. Mooij und K. Wolf. „Reducing Adapter Synthesis to Controller Synthesis“. In: *IEEE T. Services Computing* 5.1 (2012), S. 72–85 (siehe S. 9).
- [Gol99] J. S. Golan. *Semirings and Their Applications*. Kluwer Academic Publishers, 1999 (siehe S. 130).
- [Got+09] F. . Gottschalk, T. Wagemakers, M. Jansen-Vullers, W. M. P. van der Aalst und M. La Rosa. „Configurable Process Models: Experiences from a Municipality Case Study“. In: *Proceedings of the 21st International Conference on Advanced Information Systems (CAiSE 09)*. Bd. 5565. Lecture Notes in Computer Science. Berlin Heidelberg: Springer Verlag, 2009, S. 486–500 (siehe S. 9, 203).
- [Got00] K. Gottschalk. „Web Services Architecture Overview“. In: <http://www.ibm.com/developerworks/library/w-ovr/> (2000) (siehe S. 3).
- [GS10] C. Gierds und J. Sürmeli. „Estimating Costs of a Service.“ In: *ZEUS*. Hrsg. von C. Gierds und J. Sürmeli. Bd. 563. CEUR Workshop Proceedings. CEUR-WS.org, 2010, S. 121–128 (siehe S. 126).
- [Guro0] Y. Gurevich. „Sequential Abstract-State Machines Capture Sequential Algorithms“. In: *ACM Trans. Comput. Logic* 1.1 (2000), S. 77–111 (siehe S. 3).
- [HAM14] T. S. Heinze, W. Amme und S. Moser. „Message Assertions and Predicate-Based Control-Flow Unfolding Revisited“. In: *Proceedings of the 6th Central-European Workshop on Services and their Composition, ZEUS 2014, Potsdam, Germany, February 20-21, 2014*. 2014, S. 17–20 (siehe S. 224).
- [Hei14] S. Heiden. „Ein Überblick Über Programm- Und Controller-Synthese“. Studienarbeit. Humboldt-Universität zu Berlin, 2014 (siehe S. 188).
- [Hoa78] C. A. R. Hoare. „Communicating Sequential Processes“. In: *Communications of the ACM* 21.8 (1978), S. 666–677 (siehe S. 93).
- [HST14] S. Heiden, J. Sürmeli und M. Triebel. *Embedding Multi-Hemirings Into Semirings (Preprint)*. Informatik-Berichte 246. 2014 (siehe S. 129, 131–133).
- [Hul+03] R. Hull, M. Benedikt, V. Christophides und J. Su. „E-Services: A Look Behind the Curtain“. In: *the twenty-second ACM SIGMOD-SIGACT-SIGART symposium*. New York, New York, USA: ACM Press, 2003, S. 1–14 (siehe S. 3).
- [Jac+92] I. Jacobson, M. Christerson, P. Jonsson und G. Övergaard. *Object-Oriented Software Engineering - a Use Case Driven Approach*. Addison-Wesley, 1992 (siehe S. 31).
- [Jor+07] D. Jordan u. a. „Web Services Business Process Execution Language Version 2.0“. In: *OASIS Standard* 11 (2007), S. 1–264 (siehe S. 92, 209).
- [Jeh11] G. Jehle und P. Reny. *Advanced Microeconomic Theory*. The Addison-Wesley series in economics. Financial Times/Prentice Hall, 2011 (siehe S. 89).

- [Kar+94] D. Kartson, G. Balbo, S. Donatelli, G. Franceschinis und G. Conte. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, Inc., 1994 (siehe S. 224).
- [KL07] O. Kupferman und Y. Lustig. „Lattice Automata“. In: *In Proc. 8th International Conference on Verification, Model Checking, and Abstract Interpretation*. Springer, 2007, S. 199–213 (siehe S. 129).
- [Kle56] S. C. Kleene. „Representation of Events in Nerve Nets and Finite Automata“. In: *Automata Studies*. Hrsg. von C. Shannon und J. McCarthy. Princeton, NJ: Princeton University Press, 1956, S. 3–41 (siehe S. 130).
- [KM00] O. Kupferman und P. Madhusudan. „Open Systems in Reactive Environments: Control and Synthesis“. In: *Synthesis* (2000), S. 92–107 (siehe S. 8, 188).
- [KM69] R. M. Karp und R. E. Miller. „Parallel Program Schemata“. In: *Journal of Computer and System Sciences* 3.2 (1969), S. 147–195 (siehe S. 117).
- [KMR02] E. Kindler, A. Martens und W. Reisig. „Inter-Operability of Workflow Applications: Local Criteria for Global Soundness“. In: *Business Process Management*. Hrsg. von W. M. P. van der Aalst, J. Desel und A. Oberweis. Bd. 1806. Lecture Notes in Computer Science. Springer, 2002, S. 235–253 (siehe S. 32).
- [KMR97] D. Karger, R. Motwani und G. Ramkumar. „On Approximating the Longest Path in a Graph“. In: *Algorithmica* 18.1 (1997), S. 82–98 (siehe S. 202).
- [KPS06] R. Kazhamiakin, M. Pistore und L. Santuari. „Analysis of Communication Models in Web Service Compositions“. In: *the 15th international conference*. New York, New York, USA: ACM Press, 2006, S. 267–276 (siehe S. 162).
- [La 08] M. La Rosa. „Process Configuration“. In: <http://www.processconfiguration.com> (2008) (siehe S. 9, 203).
- [Lee02] E. A. Lee. „Embedded Software“. In: Elsevier, 2002, S. 55–95 (siehe S. 3).
- [LK08] N. Lohmann und J. Kleine. „Fully-Automatic Translation of Open Workflow Net Models Into Simple Abstract BPEL Processes“. In: *Modellierung 2008, 12.–14. März 2008, Berlin, Proceedings*. Hrsg. von T. Kühne, W. Reisig und F. Steimann. Bd. P-127. Lecture Notes in Informatics (LNI). GI, 2008, S. 57–72 (siehe S. 92).
- [LMW07] N. Lohmann, P. Massuthe und K. Wolf. „Operating Guidelines for Finite-State Services“. In: *Lecture Notes in Computer Science 4546*. Hrsg. von J. Kleijn und A. Yakovlev. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 321–341 (siehe S. 9, 33, 209).
- [Loh+06] N. Lohmann, P. Massuthe, C. Stahl und D. Weinberg. „Analyzing Interacting BPEL Processes“. In: *Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings*. 2006, S. 17–32 (siehe S. 209).
- [Loh+08] N. Lohmann, P. Massuthe, C. Stahl und D. Weinberg. „Analyzing Interacting Ws-Bpel Processes Using Flexible Model Generation“. In: *Data & Knowledge Engineering* 64.1 (2008), S. 38–54 (siehe S. 8, 9).
- [Loh08] N. Lohmann. „A Feature-Complete Petri Net Semantics for Ws-Bpel 2.0“. In: *Lecture Notes in Computer Science 4937*. Hrsg. von M. Dumas und R. Heckel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, S. 77–91 (siehe S. 92).
- [Loh10a] N. Lohmann. „Communication Models for Services“. In: *Proceedings of the 2nd Central-European Workshop on Services and their Composition* (2010), S. 9–16 (siehe S. 162).
- [Loh10b] N. Lohmann. „Correctness of Services and Their Composition“. Diss. Technische Universiteit Eindhoven, 2010 (siehe S. 162).

- [LW10] N. Lohmann und K. Wolf. „How to Implement a Theory of Correctness in the Area of Business Processes and Services“. In: *Lecture Notes in Computer Science* 6336. Hrsg. von R. Hull, J. Mendling und S. Tai. Springer Berlin Heidelberg, 2010, S. 61–77 (siehe S. 207).
- [LW11a] N. Lohmann und D. Weinberg. „Wendy: A Tool to Synthesize Partners for Services“. In: *Fundamenta Informaticae* 113.3 (2011), S. 295–311 (siehe S. 207).
- [LW11b] N. Lohmann und K. Wolf. „Data Under Control“. In: *Proceedings of the 18th German Workshop on Algorithms and Tools for Petri Nets (AWPN 2011), Hagen, Germany, September 29-30, 2011*. 2011, S. 34–40 (siehe S. 224).
- [Mar05] A. Martens. „Analyzing Web Service Based Business Processes“. In: *Lecture Notes in Computer Science* 3442. Hrsg. von M. Cerioli. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, S. 19–33 (siehe S. 8, 9).
- [Mas+08] P. Massuthe, A. Serebrenik, N. Sidorova und K. Wolf. „Can I Find a Partner? Undecidability of Partner Existence for Open Nets“. In: *Information Processing Letters* 108.6 (2008), S. 374–378 (siehe S. 165).
- [McI68] M. D. McIlroy. „Mass-Produced Software Components“. In: *Proceedings of the 1st International Conference on Software Engineering* (1968), S. 88–98 (siehe S. 3).
- [Men05] D. Menasce. „MOM Vs. Rpc: Communication Models for Distributed Applications“. In: *Internet Computing, IEEE* 9.2 (2005), S. 90–93 (siehe S. 162).
- [Mey88] B. Meyer. *Object-Oriented Software Construction*. Prentice Hall New York, 1988 (siehe S. 3).
- [MG11] P. Mell und T. Grance. *The Nist Definition of Cloud Computing*. Techn. Ber. 800-145. Gaithersburg, MD: National Institute of Standards und Technology (NIST), 2011 (siehe S. 3).
- [Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989 (siehe S. 33, 93).
- [Moh09] M. Mohri. „Weighted Automata Algorithms“. In: *Handbook of Weighted Automata*. Hrsg. von M. Droste, W. Kuich und H. Vogler. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2009, S. 213–254 (siehe S. 126, 129, 130, 133).
- [Moo+11] A. Mooij, J. Parnjai, C. Stahl und M. Voorhoeve. „Constructing Replaceable Services Using Operating Guidelines and Maximal Controllers“. In: *Lecture Notes in Computer Science* 6551. Hrsg. von M. Bravetti und T. Bultan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, S. 116–130 (siehe S. 9, 179).
- [Mur89] T. Murata. „Petri Nets: Properties, Analysis and Applications“. In: *Proceedings of the IEEE*. 1989, S. 541–580 (siehe S. 117).
- [Nas51] J. Nash. „Non-Cooperative Games“. In: *Annals of mathematics* (1951), S. 286–295 (siehe S. 223).
- [NNH99] F. Nielson, H. R. Nielson und C. Hankin. *Principles of Program Analysis*. Springer, 1999 (siehe S. 126).
- [Ost+12] Z. Oster, S. Ali, G. Santhanam, S. Basu und P. Roop. „A Service Composition Framework Based on Goal-Oriented Requirements Engineering, Model Checking, and Qualitative Preference Analysis“. In: *Service-Oriented Computing*. Hrsg. von C. Liu, H. Ludwig, F. Toumani und Q. Yu. Bd. 7636. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, S. 283–297 (siehe S. 203).
- [Pap07] M. M. Papazoglou. „What’s in a Service?“ In: *Software Architecture*. Lecture Notes in Computer Science 4758 (2007), S. 11–28 (siehe S. 6).
- [Pap08] M. Papazoglou. *Web Services: Principles and Technology*. Pearson Education, 2008 (siehe S. 3).

- [Pnu77] A. Pnueli. „The Temporal Logic of Programs“. In: *Foundations of Computer Science, 1977., 18th Annual Symposium on* (1977), S. 46–57 (siehe S. 3, 95).
- [Pop13] L. Popova-Zeugmann. *Time and Petri Nets*. Springer, 2013 (siehe S. 93).
- [Pos82] J. Postel. *Simple Mail Transfer Protocol*. RFC 821. RFC Editor, 1982 (siehe S. 209).
- [PR90] A. Pnueli und R. Rosner. „Distributed Reactive Systems Are Hard to Synthesize“. In: *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*. IEEE, 1990, S. 746–757 (siehe S. 8, 188).
- [Rei09] W. Reisig. „Simple Composition of Nets“. In: *Applications and Theory of Petri Nets*. Hrsg. von G. Franceschinis und K. Wolf. Bd. 5606. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, S. 23–42 (siehe S. 162).
- [Rei13] W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer, 2013 (siehe S. 3, 7, 26).
- [Rum+91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy und W. Lorensen. *Object-Oriented Modeling and Design*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991 (siehe S. 3).
- [RW87] P. Ramadge und W. Wonham. „Supervisory Control of a Class of Discrete Event Processes“. In: *SIAM Journal on Control and Optimization* 25.1 (1987), S. 206–230 (siehe S. 8).
- [Sak09] J. Sakarovitch. „Rational and Recognisable Power Series“. In: *Handbook of Weighted Automata*. Hrsg. von M. Droste, W. Kuich und H. Vogler. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2009, S. 105–174 (siehe S. 129).
- [Sal+78] A. Salomaa, M. Soittola, F. Bauer und D. Gries. *Automata-Theoretic Aspects of Formal Power Series*. Texts and monographs in computer science. Springer-Verlag, 1978 (siehe S. 129).
- [San01] D. Sangiorgi. „Asynchronous Process Calculi: The First- and Higher-Order Paradigms“. In: *Theoretical Computer Science* 253.2 (2001), S. 311–350 (siehe S. 93).
- [Sap+06] B. Sapkota, D. Roman, S. R. Kruk und D. Fensel. „Distributed Web Service Discovery Architecture“. In: *Telecommunications, 2006. AICT-ICIW'06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*. IEEE. 2006, S. 136–136 (siehe S. 9).
- [Scho3] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Algorithms and Combinatorics Bd. 1. Springer, 2003 (siehe S. 202).
- [Sch61] M. P. Schützenberger. „On the Definition of a Family of Automata“. In: *Information and Control* 4.2-3 (1961), S. 245–270 (siehe S. 129, 130).
- [Sch99] K. Schmidt. „Model-Checking with Coverability Graphs“. In: *Formal Methods in System Design* 15.3 (1999), S. 239–254 (siehe S. 117).
- [SMB09] C. Stahl, P. Massuthe und J. Bretschneider. „Deciding Substitutability of Services with Operating Guidelines“. In: *Lecture Notes in Computer Science* 5460. Hrsg. von K. Jensen und W. M. P. van der Aalst. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, S. 172–191 (siehe S. 9, 179).
- [ST13] J. Sürmeli und M. Triebel. „Synthesizing Cost-Minimal Partners for Services.“ In: *ICSOC*. Hrsg. von S. Basu, C. Pautasso, L. Zhang und X. Fu. Bd. 8274. Lecture Notes in Computer Science. Springer, 2013, S. 584–591 (siehe S. 8, 207).
- [Sta05] C. Stahl. „A Petri Net Semantics for BPEL“. In: *Informatik-Berichte* 188. Institut für Informatik, 2005 (siehe S. 92).

- [Sür12] J. Sürmeli. „Service Discovery with Cost Thresholds.“ In: *WS-FM*. Hrsg. von M. H. ter Beek und N. Lohmann. Bd. 7843. Lecture Notes in Computer Science. Springer, 2012, S. 30–48 (siehe S. 8, 39).
- [Szy02] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Pearson Education, 2002 (siehe S. 3).
- [TA10] M. Tang und L. Ai. „A Hybrid Genetic Algorithm for the Optimal Constrained Web Service Selection Problem in Web Service Composition“. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*. 2010, S. 1–8 (siehe S. 203).
- [Van+07] W. M. Van Der Aalst, M. Beisiegel, K. M. Van Hee und D. König. „An Soa-Based Architecture Framework“. In: *International Journal of Business Process Integration and Management 2.2* (2007), S. 91–101 (siehe S. 162).
- [Wag12] C. Wagner. „Partner Synthesis for Data-Dependent Services“. In: *Proceedings of the 4th Central-European Workshop on Services and their Composition, ZEUS-2012, Bamberg, Germany, February 23-24, 2012*. 2012, S. 17–24 (siehe S. 224).
- [Wei88] H. J. Weinert. „Generalized Semialgebras Over Semirings“. In: *Semigroups Theory and Applications*. Hrsg. von H. Jürgensen, G. Lallement und H. Weinert. Bd. 1320. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1988, S. 380–416 (siehe S. 21).
- [Wol+11] K. Wolf, C. Stahl, D. Weinberg, J. Ott und R. Danitz. „Guaranteeing Weak Termination in Service Discovery“. In: *Fundamenta Informaticae 108.1-2* (2011) (siehe S. 9, 143).
- [Wol07] K. Wolf. „Generating Petri Net State Spaces“. In: *Lecture Notes in Computer Science 4546*. Hrsg. von J. Kleijn und A. Yakovlev. Springer Berlin Heidelberg, 2007, S. 29–42 (siehe S. 207).
- [Wol09] K. Wolf. „Does My Service Have Partners?“ In: *Transactions on Petri Nets and Other Models of Concurrency II* (2009) (siehe S. 8, 143, 187, 202, 210).
- [Wol14] K. Wolf. „Explizites Model Checking: Welche Vorteile Bieten Petrinetze?“ In: *Informatik Spektrum 37.3* (2014), S. 220–228 (siehe S. 127).
- [Yan+14] W. Yang u. a. „A Hybrid Particle Swarm Optimization Algorithm for Service Selection Problem in the Cloud“. In: *International Journal of Grid & Distributed Computing 7.4* (2014) (siehe S. 203).
- [YL05] T. Yu und K.-J. Lin. „Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints“. In: *Service-Oriented Computing - ICSOC 2005*. Hrsg. von B. Benatallah, F. Casati und P. Traverso. Bd. 3826. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, S. 130–143 (siehe S. 94, 202).
- [Zen+04] L. Zeng u. a. „QoS-Aware Middleware for Web Services Composition“. In: *IEEE Trans. Software Eng.* 30.5 (2004), S. 311–327 (siehe S. 94, 202).

Glossar

Anforderung	Eine Menge von Systemen	86
Kostenbeschränktheit	Eine der beiden in dieser Arbeit zentralen nicht-funktionalen Verhaltens-Anforderungen	89
k-Kostenbeschränktheit	nicht-funktionale Verhaltens-Anforderung: Kosten überschreiten niemals k	89
Kostendiskriminante	Kostenschranke eines Netzes zur Entscheidung der kostenbeschränkten Bedienbarkeit	189
Kostenmodell	Festlegung der Operationen zur Aggregation von Kosten für die Definition der Kosten eines offenen Systems	46
Partner-Repräsentant	Ein Netz dessen Partnermengen für zwei Paare aus Anforderung und Präferenz übereinstimmen	175
Perspektive	Festlegung der betrachteten Abläufe für die Definition der Kosten eines offenen Systems	41
Perspektiven-Repräsentant	Ein Netz dessen Kosten unter zwei bestimmte Perspektiveen übereinstimmen	101
Präferenz	Eine Ordnung auf Systemen; größer heißt besser	91
Rahmenmarkierung	beschreibt die erlaubten Markierungen eines Netzes, wichtig für die Definition der Anforderung der schwachen Terminierung	29
Schranken-Repräsentant	Ein Netz dessen Kostenbeschränktheit mit der Markierbarkeit eines einzelnen Platzes koinzidiert	113
Schwache Terminierung	Die in dieser Arbeit zentrale funktionale Verhaltens-Anforderung	88
Wert-Repräsentant	Ein Netz dessen Gesamtkosten durch endlich viele erreichbare Markierungen bestimmt werden kann	110
Zielmarkierung	beschreibt die erwünschten Markierungen eines Netzes, wichtig für die Definition der Anforderung der schwachen Terminierung	29

Symbole

$(X)^\mu$	Der Abschluss der Menge X in μ	53
$(X)_{\leq k}^\mu$	Der k -Abschluss der Menge X in μ	54
ρ	Eine Anforderung	
\mathbb{B}	Die Menge aller Netze, die die Anforderung der Kostenbeschränktheit erfüllen	90
$\mathbb{B}[k]$	Die Menge aller Netze, die die Anforderung der k -Kostenbeschränktheit erfüllen	90
μ	Ein Kostenmodell	
$\rho[L]$	Die Menge der Partner von L bezüglich der Anforderung ρ	141
$\rho[L]^\succsim$	Die Menge der optimalen Partner von L bezüglich der Anforderung ρ und der Präferenz \succsim	154
π	Eine Perspektive	
$\text{AF}[U]$	Die Anwendungsfall-Perspektive, die nur die Transitionssequenzen erhält, die in der Sprache U^\bullet des Anwendungsfalls U vorkommen	43
$\text{FILTER}[T]$	Die Filter-Perspektive, die nur die Transitionen aus T erhält	43
$\text{TERM}[L]$	Die Terminierungs-Perspektive von L	43
$\text{WFILTER}[w]$	Die Wort-Filter-Perspektive, die nur die Transitionssequenzen erhält, die das Wort w erzeugen	43
$\text{ZF}[N]$	Die Perspektive, die nur die Transitionssequenzen von N erhält, die keinen Zyklus bilden	43
\succsim	Eine Präferenz	
\mathbb{S}	Die Menge aller Netze, die die Anforderung der schwachen Terminierung erfüllen	89

Danksagung

Als erstes möchte ich Wolfgang Reisig dafür danken, dass er mich in seine Forschungsgruppe aufnahm, mich in die Welt der Wissenschaft einführte, mir beibrachte, Vorträge zu halten, und mich bei meiner Promotion unterstützte. Ich danke außerdem Louchka Popova-Zeugmann und Manfred Droste für die Zeit, die sie sich für Betreuung und Begutachtung nahmen. Nicole Schweikardt, Uwe Nestmann und Klaus Ahrens danke ich zudem für ihre Tätigkeit in meiner Promotionskommission.

In den letzten Jahren genoss ich teils täglich die Gesellschaft einiger Menschen, denen ich zu ganz besonderem Dank verpflichtet bin: Birgit Heene, Hartmut Lackner, Christian Gierds, Robert Prüfer, Richard Müller, Andre Moelle, Kim Völlinger, Raffael Dzikowski, Marvin Triebel und Simon Heiden. Danke für Eure jahrelange Unterstützung und die vielen guten gemeinsamen Stunden, die weit mehr als schnöder Arbeitsalltag waren. Daniela Weinberg, Peter Massuthe, Karsten Wolf, Dirk Fahland, Niels Lohmann und Christian Stahl – auch Euch möchte ich für die gute Zusammenarbeit und die vielen guten Denkanstöße danken. Vielen Dank auch an die Mitglieder der B.E.S.T.-Kooperation und des Graduiertenkollegs SOAMED – ich habe mich dort stets sehr wohlfühlt.

Zum Schluss möchte ich meiner Familie und meinen Freunden für die vielen schönen Momente danken, mit denen sie mein Leben bereicherten, mir stets den notwendigen Rückhalt gaben, und damit einen nicht unerheblichen Teil zu meiner Promotion beitrugen.